



UNIVERSIDAD CARLOS III DE MADRID

TESIS DOCTORAL

Contribución al reconocimiento de huellas dactilares mediante técnicas de correlación y arquitecturas hardware para el aumento de prestaciones

Autor:

Almudena Lindoso Muñoz

Director:

Dr. Luis Alfonso Entrena Arrontes

DEPARTAMENTO DE TECNOLOGÍA ELECTRÓNICA

Leganés, Febrero 2009

TESIS DOCTORAL

CONTRIBUCIÓN AL RECONOCIMIENTO DE HUELLAS DACTILARES MEDIANTE TÉCNICAS DE CORRELACIÓN Y ARQUITECTURAS HARDWARE PARA EL AUMENTO DE PRESTACIONES

Autor: Almudena Lindoso Muñoz

Director: Dr. Luis Alfonso Entrena Arrontes

Firma del Tribunal Calificador:

Presidente: **Dr. Emilio Olías Ruiz**

Firma

Vocal: **Dra. Teresa Riesgo Alcaide**

Vocal: **Dr. José Javier Garrido Salas**

Vocal: **Dr. Juan Carlos Pérez Cortés**

Secretario: **Dr. Miguel Ángel Aguirre Echanove**

Calificación:

Leganés, de de

A mis padres

A Kim

AGRADECIMIENTOS

En primer lugar quiero agradecer a Luis Entrena el trabajo y el tiempo que ha dedicado a esta tesis doctoral. Quiero agradecerle su apoyo y optimismo en todo momento, porque incluso cuando las ideas, resultados o revisiones no eran favorables, él siempre ha sabido encontrar el lado positivo. Sin su ayuda, su visión crítica y su guía, esta tesis doctoral no habría sido posible. Muchas gracias por confiar en mí, en mis ideas y en definitiva por dirigir esta tesis.

También quiero agradecer la ayuda y el apoyo de todos mis compañeros del Departamento de Tecnología Electrónica. En especial a mis compañeros del grupo de Microelectrónica, a todos los que fueron y los que son. Gracias a todos por ayudarme y aconsejarme a manejar a los alumnos, las clases y la investigación sin volverme loca en el intento. Muchas gracias a Mario por sus numerosos rescates informáticos que me han salvado de un desastre más de una vez. En este apartado también tengo que agradecer a Kim sus rescates, sobre todo el último, recuperando todos los datos de un disco duro que decidió romperse en el peor momento. Muchas gracias a Enrique por guiarnos en todos los complejos procesos de papeleo de esta universidad, sin él habríamos tardado siglos en hacer todos los papeles (no se lo que vamos a hacer Judith, Juan Carlos y yo ahora que te has adelantado). También quiero agradecer su apoyo en este largo camino a las chicas de Microelectrónica: Celia, Susana, Judith y Marta. Muchas gracias por todo durante todos estos años, creo que ya va siendo hora de que el comando Gambetta repita misión. Muchas gracias a Judith por aportar una visión crítica y práctica en cualquier ámbito. Sin tu

compañía, consejos y apoyo a lo largo de esta tesis, el camino habría sido muchísimo más duro.

No quiero olvidar a mis compañeros del antiguo y enorme despacho, todos los que pasaron por allí han compartido conmigo esta tesis. Aunque mis nuevos compañeros de despacho son estupendos y han sufrido el final de esta tesis, siempre recordaré con cariño el antiguo despacho, enorme y ruidoso. Gracias por todos los buenos momentos que hemos compartido durante estos años. Espero que en breve estéis todos escribiendo y defendiendo vuestras tesis.

Y en el ámbito personal creo que esta tesis tampoco habría sido posible sin el apoyo incondicional de Kim y su confianza en mí. Gracias por ser mi escolta por medio mundo, alegrarte de los resultados positivos y animarme con los menos positivos.

RESUMEN

Las huellas dactilares son un identificador biométrico ampliamente utilizado. Su uso se extiende desde aplicaciones policiales y forenses hasta aplicaciones civiles muy comunes, como el control de accesos. Con la aparición de los primeros sistemas automáticos de reconocimiento de huellas (AFIS, “Automatic Fingerprint Identification System”) se comenzó el desarrollo y utilización de algoritmos de comparación de huellas dactilares basados en minucias, que son puntos singulares dentro de la huella. Estos métodos se asemejan a la comparación visual realizada por un experto, identificando la posición de los puntos singulares para su posterior comparación. Los métodos de comparación basados en minucias proporcionan una respuesta precisa pero presentan numerosos inconvenientes. En primer lugar, las técnicas de comparación basadas en minucias requieren un fuerte preprocesado de las imágenes que suelen introducir artefactos en las imágenes a comparar. En segundo lugar, estas técnicas solo toman una parte muy concreta de la información contenida en la huella, sesgando con ello en gran medida la comparación. Actualmente los sensores de huellas dactilares son capaces de proporcionar imágenes de alta resolución y con alta calidad, utilizar únicamente las minucias para la comparación es desaprovechar el resto de la valiosa información contenida en la huella.

Los algoritmos basados en técnicas de correlación se presentan como candidatos para paliar los dos inconvenientes anteriormente mencionados. Estas técnicas utilizan el cálculo de la correlación cruzada como medida de similitud de las imágenes. Por lo tanto se compara toda la información contenida en la huella y

además, como no se requiere la extracción de características, se evita la compleja etapa de preprocesado requerida por los métodos basados en minucias. A pesar de haber demostrado recientemente que con las técnicas basadas en correlación se pueden obtener resultados precisos [FVC2002], [FVC2004], estas técnicas siguen siendo minoritariamente estudiadas y utilizadas. La principal desventaja que presentan es su alta carga computacional, requiriendo equipos de altas prestaciones, coste y tamaño.

Esta tesis doctoral presenta algoritmos para la comparación de huellas dactilares mediante técnicas de correlación así como arquitecturas hardware eficientes para solventar los problemas de rendimiento que presentan estas técnicas.

Se han propuesto dos algoritmos de comparación de huellas dactilares mediante técnicas de correlación que son aportación original de esta tesis. Los algoritmos presentan dos soluciones diferenciadas según la resolución de las imágenes a comparar. Los métodos de alineamiento propuestos para cada uno de los algoritmos también utilizan técnicas novedosas y son por tanto aportación original de esta tesis.

El algoritmo de comparación de imágenes de baja resolución realiza una etapa de alineamiento novedosa basada en la correlación del campo de orientación de las huellas. A partir del desplazamiento y la rotación calculados se seleccionan las zonas de las huellas a comparar teniendo en cuenta el solape que existe entre las huellas así como la calidad de las zonas elegidas. Una vez determinadas las zonas se utiliza la correlación cruzada como medida de similitud de las zonas.

El algoritmo de comparación para imágenes de alta resolución selecciona zonas cercanas al núcleo para efectuar la comparación. La selección del núcleo se efectúa mediante una técnica novedosa que utiliza la transformada wavelet para determinar la posición del núcleo. La detección del núcleo se realiza sobre la imagen directamente sin necesidad de aplicar ningún preprocesado previo. Una vez seleccionadas las zonas que contienen al núcleo, se realiza el cálculo de la correlación cruzada como medida de similitud de las imágenes.

Los resultados experimentales para ambos algoritmos determinan que ambos métodos son precisos en la comparación pero su rendimiento en implementaciones

software es bajo. Esto es debido a que el rendimiento en ambos algoritmos se ve afectado por la elevada carga computacional que acarrea el cálculo de la correlación cruzada.

En esta tesis también se han propuesto arquitecturas hardware para la aceleración del cálculo de la correlación cruzada. En concreto, se han propuesto dos arquitecturas hardware, una para el cálculo de la correlación en el dominio espacial y otra para el cálculo de la correlación en el dominio espectral. Así como una arquitectura para realizar por hardware la normalización y poder así obtener una correlación normalizada. Todas las arquitecturas propuestas son aportación original de esta tesis.

La arquitectura espacial consta de una matriz sistólica de DSP slices que realizan MACs (“Multiplication ACcumulation”). En esta arquitectura cada DSP slice realiza una MAC y pasa el resultado al siguiente DSP slice de su misma fila. Por lo tanto, en cada fila de la matriz se calcula la correlación cruzada de una fila de ambas imágenes. Después de un retardo inicial, los resultados se proporcionan cada ciclo de reloj.

La arquitectura espectral se basa en la aplicación del teorema de la correlación y en la utilización de la transformada rápida de Fourier (FFT). Se han utilizado módulos de cálculo de la FFT, que se han denominado FFT slices. En la arquitectura propuesta, cada FFT slice realiza la FFT de una fila de ambas imágenes. La arquitectura gestiona los FFT slices, así como memorias intermedias, para realizar el máximo número posible de FFTs en paralelo.

Para la normalización se presentan arquitecturas recursivas que aprovechan el flujo de datos de las arquitecturas que calculan la correlación. Con esta aproximación es posible combinar ambos circuitos para integrarlos en un solo chip. Además de esto, al compartir el flujo de datos los resultados se generan a la vez, sin disminuir el rendimiento del sistema.

Experimentalmente se ha comprobado la aceleración conseguida con las arquitecturas propuestas mediante implementaciones de las arquitecturas en una FPGA (“Field Programmable Gate Array”) de la familia Virtex 4 de Xilinx

[Xil_V4]. Los resultados experimentales demuestran que con las arquitecturas propuestas se pueden conseguir aceleraciones de hasta dos órdenes de magnitud respecto a implementaciones en C en un PC de gama alta.

También se han propuesto soluciones a nivel de sistema para la comparación de huellas dactilares mediante técnicas de correlación. Estas soluciones presentan la integración de las arquitecturas hardware propuestas en un SoPC (“System On Pogrammable Chip”). La arquitectura básica del SoPC propuesto consta de un microprocesador empotrado junto con un coprocesador hardware que realiza las tareas de complejidad elevada. Con esta arquitectura se busca una mejora en el rendimiento del sistema, aliviando la carga computacional del microprocesador empotrado.

Se proponen dos arquitecturas para el coprocesador hardware del SoPC que son aportación original de esta tesis. La arquitectura estática se basa en la arquitectura espacial propuesta, pero utilizando una única fila de DSP slices para realizar los cálculos. En esta arquitectura, el coprocesador se encarga de realizar las tareas de mayor carga computacional y el microprocesador el resto de tareas necesarias para la comparación, incluyendo el envío de datos al coprocesador y la recomposición de resultados.

La arquitectura dinámica se compone de un coprocesador reconfigurable de grano grueso que puede cambiar en tiempo de ejecución tanto la operación a realizar como el tamaño de las imágenes empleadas, implicando un redimensionamiento de la matriz de DSP slices del coprocesador. La reconfiguración la realiza directamente el microprocesador empotrado, sin necesidad de utilizar los mecanismos internos de reconfiguración de la FPGA. Este coprocesador presenta una solución versátil a la par que eficiente. La pérdida de posibilidades frente a una reconfiguración de grano fino se ve compensada por una disminución drástica del tiempo de reconfiguración que con la arquitectura propuesta puede realizarse con tan solo cuatro transacciones de 32 bits.

El sistema se ha prototipado para una FPGA Virtex 5 de Xilinx [Xil_V5] utilizando como microprocesador empotrado Microblaze [Xil_mblaze]. Los

resultados experimentales obtenidos muestran que el SoPC diseñado es altamente eficiente para diversas operaciones comunes en el campo del procesado de imágenes (CC, S, SS y transformada wavelet) superando en rendimiento a implementaciones software en un PC de altas prestaciones. En el SoPC propuesto también se ha implementado el algoritmo para imágenes de alta resolución consiguiendo el mismo rendimiento que un PC de altas prestaciones.

Con esta tesis doctoral se ha estudiado la comparación de huellas dactilares mediante técnicas de correlación obteniendo resultados precisos con algoritmos novedosos de comparación. Se ha demostrado que las técnicas de correlación son unas técnicas prometedoras en el ámbito de la comparación de huellas dactilares, por la precisión obtenida así como por su capacidad de comparar la totalidad de la información contenida en la huella.

También se han solventado los problemas de rendimiento que presentan estas técnicas mediante arquitecturas hardware específicas que proporcionan rendimientos elevados. Las arquitecturas propuestas hacen posible cálculos de correlación en tiempo real y en sistemas de bajo coste y tamaño. El SoPC con coprocesador dinámico propuesto presenta una solución muy versátil, ya que es capaz de modificar en tiempo de ejecución la tarea a realizar por el coprocesador, el tamaño de las imágenes y de la matriz de DSP slices, realizando el cálculo de forma eficiente para un conjunto de datos de gran tamaño.

ÍNDICE

ÍNDICE	1
ÍNDICE DE FIGURAS.....	4
ÍNDICE DE TABLAS.....	6
CAPÍTULO 1: INTRODUCCIÓN.....	9
1.1- MOTIVACIÓN	10
1.2- OBJETIVOS.....	13
1.3- ESTRUCTURA DE LA TESIS.	14
CAPÍTULO 2: RECONOCIMIENTO MEDIANTE HUELLA DACTILAR Y ACELERACIÓN HARDWARE	17
2.1- INTRODUCCIÓN.....	18
2.2- LA HUELLA DACTILAR COMO IDENTIFICADOR BIOMÉTRICO.....	19
2.2.1- <i>Sistemas de reconocimiento mediante huella dactilar</i>	22
2.2.1.2- Métricas de evaluación.....	24
2.2.2- <i>Algoritmos de comparación mediante huellas dactilares</i>	26
2.2.2.1- Técnicas basadas en minucias	27
2.2.2.2- Técnicas basadas en crestas	29
2.2.2.3- Técnicas de correlación.....	30
2.3- AUMENTO DE PRESTACIONES MEDIANTE ACELERACIÓN HARDWARE	34
2.4- CONCLUSIONES	40
CAPÍTULO 3: COMPARACIÓN DE HUELLAS DACTILARES MEDIANTE TÉCNICAS DE CORRELACIÓN.....	45
3.1- INTRODUCCIÓN.....	46
3.2- INTRODUCCIÓN A LA CORRELACIÓN DE IMÁGENES	49
3.3- ALGORITMO DE COMPARACIÓN PARA IMÁGENES DE BAJA RESOLUCIÓN	52
3.2.1- <i>Preprocesado</i>	53
3.2.2- <i>Alineamiento</i>	56

3.2.3-	<i>Selección de zonas</i>	58
3.2.4-	<i>Comparación</i>	60
3.4-	ALGORITMO DE COMPARACIÓN PARA IMÁGENES DE ALTA RESOLUCIÓN	62
3.3.1-	<i>Selección de la información dentro de la huella</i>	63
3.3.2-	<i>Selección de zonas</i>	70
3.3.3-	<i>Comparación</i>	76
3.4-	RESULTADOS EXPERIMENTALES	78
3.4.1-	<i>Algoritmo de comparación para huellas de baja resolución</i>	78
3.4.2-	<i>Algoritmo de comparación para huellas de alta resolución</i>	81
3.4.2.1-	Detección del núcleo	81
3.4.2.2-	Algoritmo de comparación	85
3.5-	CONCLUSIONES	88
CAPÍTULO 4: ACELERACIÓN HARDWARE DEL CÁLCULO DE LA CORRELACIÓN		91
4.1-	INTRODUCCIÓN	92
4.2-	CORRELACIÓN CRUZADA	94
4.3-	ARQUITECTURA ESPACIAL	98
4.4-	ARQUITECTURA ESPECTRAL	104
4.5-	NORMALIZACIÓN Y COMPARACIÓN	107
4.6-	RESULTADOS EXPERIMENTALES	110
4.6.1-	<i>Implementación de las arquitecturas y selección de FPGA</i>	111
4.6.2-	<i>Consumo de recursos y rendimiento</i>	113
4.6.3-	<i>Aplicación a la comparación de huellas dactilares</i>	116
4.7-	CONCLUSIONES	125
CAPÍTULO 5: ARQUITECTURAS HARDWARE PARA LA COMPARACIÓN EFICIENTE DE HUELLAS DACTILARES		129
5.1-	INTRODUCCIÓN	130
5.2-	DESCRIPCIÓN DE LA ARQUITECTURA DEL SISTEMA	132
5.3-	COMPONENTES DEL SISTEMA	134
5.3.1-	<i>FPGA</i>	134
5.3.2-	<i>Microprocesador</i>	135
5.3.3-	<i>Comunicaciones</i>	136
5.4-	COPROCESADOR HARDWARE	139
5.4.1-	<i>Arquitectura estática</i>	140
5.4.2-	<i>Arquitectura dinámica</i>	142
5.5-	ALGORITMO IMPLEMENTADO	147
5.5.1-	<i>Transformada wavelet</i>	148
5.5.2-	<i>ZNCC</i>	149

5.6-	SOFTWARE.....	151
5.7-	RESULTADOS EXPERIMENTALES	152
5.7.1-	<i>Precisión</i>	153
5.7.2-	<i>Consumo de recursos</i>	154
5.7.3-	<i>Rendimiento</i>	156
5.8-	CONCLUSIONES	162
CAPÍTULO 6:	CONCLUSIONES.....	165
6.1-	CONCLUSIONES	166
6.2-	LÍNEAS FUTURAS	172
BIBLIOGRAFÍA	175
ACRÓNIMOS	197

ÍNDICE DE FIGURAS

FIGURA 2.1. HUELLA DACTILAR.....	21
FIGURA 2.2.ESQUEMA TÍPICO DE AFIS.....	22
FIGURA 2.3 TASAS DE ERROR DE UN SISTEMA BIOMÉTRICO.	25
FIGURA 3.1. DIAGRAMA DE BLOQUES DEL ALGORITMO DE COMPARACIÓN.....	53
FIGURA 3.2. HUELLA PREPROCESADA	54
FIGURA 3.3. HUELLA DE ENTRADA Y SU CAMPO DE ORIENTACIÓN	55
FIGURA 3.4. SELECCIÓN DE ZONAS.....	60
FIGURA 3.5. COMPARACIÓN LOCAL	61
FIGURA 3.6. TRANSFORMADA WAVELET NIVEL 1 Y 2.....	64
FIGURA 3.7. SELECCIÓN DE ZONAS DB PROPIA	67
FIGURA 3.8. SELECCIÓN DE ZONAS FVC 2006(1)	67
FIGURA 3.9. SELECCIÓN DE ZONAS FVC 2006(2)	68
FIGURA 3.10. SEGMENTACIÓN DB PROPIA.....	69
FIGURA 3.11. SEGMENTACIÓN FVC 2006	70
FIGURA 3.12. ORIENTACIÓN PREDOMINANTE (H/V).....	72
FIGURA 3.13. CÁLCULO DE LA COMPONENTE PREDOMINANTE H/V	73
FIGURA 3.14. COMPONENTE PREDOMINANTE H/V (FVC 2006).....	74
FIGURA 3.15. PROCESO PARA LA DETECCIÓN DEL NÚCLEO.....	75
FIGURA 3.16. DETECCIÓN DEL NÚCLEO (FVC 2006).....	76
FIGURA 3.17. ÁREA DE BÚSQUEDA Y POSICIÓN DEL NÚCLEO	77
FIGURA 3.18. ROC EN ESCALA LOGARÍTMICA PARA EL ALGORITMO DE COMPARACIÓN PARA IMÁGENES DE BAJA RESOLUCIÓN PARA LA BASE DE DATOS FVC 2000 2 A.....	79
FIGURA 3.19. HUELLAS DE BAJA CALIDAD	80

FIGURA 3.20. ERRORES DE DETECCIÓN DEL NÚCLEO: (1) HUELLAS CON BAJA CALIDAD, (2) NÚCLEO EN EL EXTREMO SUPERIOR DE LA IMAGEN, (3) HUELLAS TIPO ARCO CON BAJA CURVATURA	84
FIGURA 3.21. ROC EN ESCALA LOGARÍTMICA PARA EL ALGORITMO DE ALTA RESOLUCIÓN PARA LA BASE DE DATOS DE CALIBRACIÓN DEL FVC2006. (120 HUELLAS).....	86
FIGURA 3.22. ROC EN ESCALA LOGARÍTMICA PARA EL ALGORITMO DE ALTA RESOLUCIÓN PARA DB2 FVC2006. (1680 HUELLAS)	87
FIGURA 4.1. BLOQUE DSP SLICE DE XILINX PARA LA FAMILIA VIRTEX-4	99
FIGURA 4.2. ARQUITECTURA ESPACIAL PARA EL CÁLCULO DE LA CORRELACIÓN CRUZADA	100
FIGURA 4.3. INTERCONEXIÓN DE DSP SLICES DENTRO DE UNA MISMA FILA ..	101
FIGURA 4.4. ARQUITECTURA ESPECTRAL PARA EL CÁLCULO DE LA CORRELACIÓN CRUZADA	105
FIGURA 4.5. ORGANIZACIÓN DEL ACCESO A LAS MEMORIAS	107
FIGURA 4.6. ARQUITECTURA PARA EL CÁLCULO DE S(P, Q).....	109
FIGURA 4.7. TIEMPO DE PROCESADO VERSUS TAMAÑO DE LAS REGIONES	121
FIGURA 5.1. ARQUITECTURA DEL SISTEMA	133
FIGURA 5.2. ARQUITECTURA DE MICROBLAZE.....	135
FIGURA 5.3. COPROCESADOR HARDWARE (CC)	141
FIGURA 5.4. COPROCESADOR HARDWARE (S Y SS).....	141
FIGURA 5.5. ESQUEMA DE CÁLCULO DE S Y SS POR FILAS	142
FIGURA 5.6. ARQUITECTURA DEL COPROCESADOR.....	144
FIGURA 5.7. ARQUITECTURA DE LA MATRIZ DE DSP SLICES.....	146
FIGURA 5.8. ARQUITECTURA SIMPLIFICADA DE UN DSP SLICE.....	146
FIGURA 5.9. ESQUEMA DEL CÓDIGO EJECUTADO POR EL MICROPROCESADOR EMPOTRADO Y LAS PARTES REALIZADAS POR EL COPROCESADOR HARDWARE	151

ÍNDICE DE TABLAS

TABLA 3.1. ERRORES EN LA LOCALIZACIÓN DEL NÚCLEO PARA LA FVC2006 DB2 (1680 HUELLAS)	82
TABLA 4.1. RESULTADOS EXPERIMENTALES (CONSUMO DE RECURSOS DE LA FPGA)	113
TABLA 4.2. RESULTADOS EXPERIMENTALES (RENDIMIENTO)	114
TABLA 4.3. RENDIMIENTO DE LAS IMPLEMENTACIONES SOFTWARE DE LA COMPARACIÓN DE HUELLAS DACTILARES BASADA EN TÉCNICAS DE CORRELACIÓN.....	118
TABLA 4.4. RENDIMIENTO DE IMPLEMENTACIONES HARDWARE DE LA COMPARACIÓN DE HUELLAS DACTILARES BASADA EN TÉCNICAS DE CORRELACIÓN.....	120
TABLA 4.5. RENDIMIENTO DE IMPLEMENTACIONES HARDWARE DE LOS ALGORITMOS DE BAJA Y ALTA RESOLUCIÓN.....	124
TABLA 5.1. CONSUMO DE RECURSOS DEL SOPC	154
TABLA 5.2. CONSUMO DE RECURSOS DEL COPROCESADOR RECONFIGURABLE	155
TABLA 5.3. MÁXIMA FRECUENCIA DEL COPROCESADOR RECONFIGURABLE.	155
TABLA 5.4. RENDIMIENTO DEL MODO DE OPERACIÓN TRANSFORMADA WAVELET	156
TABLA 5.5. RENDIMIENTO DEL MODO DE OPERACIÓN CC.....	157
TABLA 5.6. RENDIMIENTO DEL MODO DE OPERACIÓN S	158
TABLA 5.7. RENDIMIENTO DEL MODO DE OPERACIÓN SS	158
TABLA 5.8. RENDIMIENTO DEL ALGORITMO DE VERIFICACIÓN DE HUELLAS DACTILARES (PATRÓN DE 48X48)	159
TABLA 5.9. RENDIMIENTO DEL ALGORITMO DE VERIFICACIÓN DE HUELLAS DACTILARES (PATRÓN DE 24X24)	161
TABLA 5.10. RENDIMIENTO DE LA ETAPA DE COMPARACIÓN EN EL SOPC.....	162

Capítulo 1:

INTRODUCCIÓN

1.1-	MOTIVACIÓN	10
1.2-	OBJETIVOS.....	13
1.3-	ESTRUCTURA DE LA TESIS.	14

1.1- Motivación

En la actualidad la huella dactilar es un identificador biométrico profusamente utilizado tanto en los ámbitos forense y policial como en el ámbito civil. Los fundamentos del reconocimiento de huellas dactilares y su adopción para el uso forense datan de finales del siglo XIX. Su uso se extendió rápidamente y desde entonces se ha venido utilizando de forma rutinaria como medio de identificación. A partir de los años sesenta del pasado siglo se empiezan a desarrollar sistemas de reconocimiento automático de huellas dactilares, conocidos como AFIS (“Automatic Fingerprint Identification Systems”), cuyo uso ha permitido extender la utilización de la huella dactilar como medio de identificación para un gran número de aplicaciones, incluyendo aplicaciones civiles.

En un principio las técnicas de reconocimiento automático de las huellas dactilares trataron de emular las técnicas que los expertos utilizaban manualmente para analizar una huella dactilar. Estas técnicas se basan en encontrar los puntos singulares, denominados minucias, y su posición relativa dentro de la huella para poder compararlos. Este tipo de técnicas, denominadas técnicas basadas en minucias, son las más estudiadas y utilizadas pero también presentan una serie de problemas conocidos.

En primer lugar, la extracción y comparación de minucias no es una tarea trivial, ya que es necesario un fuerte preprocesado de la imagen de la huella dactilar para poder extraer los puntos singulares. En el proceso, la imagen inicialmente capturada cambia enormemente y se puede ver afectada por el ruido introducido en los sucesivos filtrados. Este proceso produce un mapa de puntos singulares de la huella, que será comparado con otro mapa de puntos singulares de otra huella, y con el resultado se toma una decisión sobre si las dos muestras pertenecen al mismo dedo. El ruido introducido en el preprocesado influye en la extracción de las minucias, generando falsas minucias o cambiando su posición relativa, dificultando con ello el proceso de comparación. En segundo lugar, las técnicas basadas en minucias se

basan en extraer los puntos singulares para comparar su posición y características, sin considerar el resto de la información contenida en la huella. La huella dactilar es una textura que contiene valiosa información que puede ser utilizada en el proceso de comparación, y utilizar únicamente la posición de sus puntos singulares es sesgar en gran medida la información contenida en ella.

Hoy en día, los métodos de captura de las huellas dactilares han avanzado considerablemente, proporcionando imágenes de muy alta calidad e incluso sin necesidad de contacto [Parziale_06]. Esta evolución no ha tenido todavía importantes consecuencias en el campo de las técnicas de comparación, ya que por el momento se siguen utilizando las técnicas basadas en minucias mayoritariamente. Sin embargo, las imágenes que pueden producir los nuevos sensores necesitan nuevos métodos que sean capaces de comparar toda la información contenida en ellas y explotarla para realizar un reconocimiento más robusto, sin necesidad de reducirla a un subconjunto limitado de características como pueden ser las minucias.

Recientemente las técnicas de comparación de huellas dactilares basadas en correlación han destacado por sus buenos resultados en las competiciones de comparación de huellas FVC (“Fingerprint Verification Competition”) [FVC2002], [FVC2004]. Estas técnicas utilizan la correlación de las huellas dactilares para medir su similitud y poder emitir un veredicto. Teóricamente, en este caso no es necesario realizar una extracción de características ya que las imágenes pueden ser comparadas directamente. Por tanto, el preprocesado requerido para estas técnicas disminuye considerablemente ya que no se buscan los puntos singulares de las crestas. Por otra parte, la correlación permite considerar toda la información contenida en la huella.

Sin embargo, las técnicas basadas en correlación también presentan dificultades, siendo la principal de ellas que el cálculo de la correlación conlleva una alta complejidad computacional. Además, en este caso, la complejidad de la operación realizada es directamente proporcional a la resolución de las imágenes utilizadas. El coste computacional lleva necesariamente asociado un incremento en el tiempo necesario para la comparación. Estos efectos hacen necesaria la utilización de

procesadores de gama alta para poder alcanzar una eficiencia razonable, lo que implica a su vez un incremento en el coste en cualquier sistema comercial. Aun así, el tamaño de las zonas de correlación o las posibilidades de considerar traslaciones y rotaciones entre las muestras están generalmente muy limitados si se quiere obtener una respuesta en un tiempo aceptable. En consecuencia, y a pesar de su teórica relevancia, estas técnicas han recibido tradicionalmente poca atención y su estudio ha sido minoritario.

Por las razones expuestas anteriormente, el estudio y mejora de las técnicas de comparación de huellas dactilares mediante técnicas de correlación reviste un gran interés, particularmente en atención a las más modernas generaciones de sensores que proporcionan una mayor resolución y una mayor calidad de las imágenes. Para estos casos es posible realizar una comparación más precisa si los sistemas de reconocimiento consideran toda la información disponible en la imagen de la huella, sin limitarse exclusivamente a los puntos singulares. Sin embargo, los métodos que permiten considerar características adicionales están muy poco desarrollados. Entre estos métodos se encuentran las técnicas de correlación.

Si el principal inconveniente de las técnicas de correlación son sus prestaciones, una forma de mejorarlas consiste en utilizar arquitecturas hardware a medida que aceleren el procesamiento. Estas arquitecturas pueden contribuir a un incremento de la precisión de los algoritmos de comparación, ya que con ellas sería posible realizar más correlaciones en un menor tiempo. Por otra parte, la aceleración hardware puede realizarse con un bajo coste sobre tecnologías reconfigurables, que vienen teniendo un desarrollo muy notable en los últimos años. En este sentido, cabe destacar que las actuales FPGAs (“Field Programmable Gate Array”) contienen recursos hardware muy útiles y eficientes para realizar tareas de procesamiento digital de señal, tales como multiplicadores, bloques de memoria, etc. Estos recursos favorecen particularmente la aceleración de algoritmos que presenten una alta regularidad de sus flujos de datos, como es el caso de los algoritmos de correlación.

1.2- Objetivos

El objetivo fundamental de esta tesis es desarrollar algoritmos de comparación de huellas dactilares basados en técnicas de correlación, así como proponer arquitecturas hardware que permitan mejorar sus prestaciones. De forma más detallada, los objetivos de esta tesis se indican a continuación:

- Proponer, diseñar y caracterizar algoritmos de comparación basados en técnicas de correlación para huella dactilar, analizando su problemática específica y aportando soluciones que mejoren su precisión y su eficiencia.
- Proponer y diseñar arquitecturas hardware para la aceleración de algoritmos de comparación de huellas dactilares basados en técnicas de correlación, con especial atención a su implementación sobre dispositivos programables, (FPGAs).
- Estudiar la integración de las arquitecturas hardware propuestas dentro de un sistema empotrado. Proponer y diseñar sistemas empotrados eficientes que soporten la aceleración hardware de algoritmos de huellas dactilares basados en técnicas de correlación.
- Explorar la capacidad de reconfiguración de las FPGAs para adaptarlas al diseño de algoritmos de comparación de huellas dactilares mediante técnicas de correlación.

En el desarrollo de estos objetivos, es conveniente destacar que se han considerado dos premisas importantes. En primer lugar, reducir al mínimo la utilización de algoritmos de preprocesado con el fin de determinar la eficacia de las técnicas de correlación por sí mismas. Es evidente que un correcto preprocesado proporciona generalmente una mayor calidad de imagen y por tanto favorece la comparación, pero también puede introducir artefactos que afectan negativamente a

dicha comparación. En consecuencia, la caracterización de los algoritmos propuestos se ha realizado directamente sobre los mismos, sin perjuicio de que los resultados puedan mejorarse mediante una adecuada combinación con otras técnicas, entre las cuales figuraría obviamente la consideración de un preprocesado más exhaustivo.

En segundo lugar, en el diseño de los algoritmos se ha tenido siempre presente la utilización de técnicas que presenten una alta regularidad, de manera que se favorezca una implementación hardware o el reaprovechamiento de recursos hardware en diferentes fases del procesamiento.

1.3- Estructura de la tesis.

El documento de esta tesis se ha dividido en seis capítulos. En el primer capítulo se resumen la motivación y los objetivos de esta tesis.

En el segundo capítulo se resume el estado de la técnica tanto en el campo del reconocimiento de la huella dactilar como en el de las tecnologías reconfigurables, centrándose este último en el fabricante de FPGAs Xilinx [Xilinx] que es el que se ha utilizado para el desarrollo de la parte experimental de esta tesis.

En el tercer capítulo se presentan dos algoritmos de comparación para el reconocimiento mediante huellas dactilares basados en las técnicas de correlación. Estas dos soluciones están adaptadas a la resolución que tengan las huellas dactilares. Cada uno de los algoritmos propone una técnica de alineamiento novedosa adecuada a la comparación de huellas dactilares basada en técnicas de correlación.

En el cuarto capítulo se presentan arquitecturas hardware para la aceleración del reconocimiento mediante huellas dactilares basado en técnicas de correlación. Las arquitecturas propuestas en este capítulo realizan el cálculo de la correlación en el dominio espacial y en el espectral así como la normalización de resultados.

En el quinto capítulo se presentan arquitecturas de sistema en dispositivos reconfigurables para la comparación de huellas dactilares mediante técnicas de correlación. A diferencia con el cuarto capítulo, en este capítulo las arquitecturas incluyen un microprocesador empotrado, mejorando con ello la gestión de las comunicaciones y el desempeño de ciertas tareas necesarias para la comparación. La otra pieza clave de los sistemas presentados es el coprocesador hardware, que se encargará de las tareas con mayor carga computacional para aliviar la carga soportada por el microprocesador empotrado y aumentar con ello el rendimiento global del sistema. En este capítulo se presentan dos SoPC (“System on Programmable Chip”) que se diferencian en el coprocesador hardware utilizado: coprocesador estático y coprocesador dinámico. El coprocesador dinámico aprovechará la capacidad intrínseca de reconfiguración de la FPGA en la que se prototipa el SoPC para aumentar las prestaciones del sistema mediante arquitecturas de grano grueso reconfigurables.

En el sexto capítulo se enumeran las principales conclusiones obtenidas a partir de esta Tesis, resumiendo los resultados obtenidos en cada capítulo y proponiendo posibles líneas futuras a partir del trabajo realizado.

Capítulo 2:

RECONOCIMIENTO MEDIANTE HUELLA DACTILAR Y ACELERACIÓN HARDWARE

2.1-	INTRODUCCIÓN.....	18
2.2-	LA HUELLA DACTILAR COMO IDENTIFICADOR BIOMÉTRICO.....	19
2.2.1-	<i>Sistemas de reconocimiento mediante huella dactilar.....</i>	22
2.2.2-	<i>Algoritmos de comparación mediante huellas dactilares.....</i>	26
2.3-	AUMENTO DE PRESTACIONES MEDIANTE ACELERACIÓN HARDWARE	34
2.4-	CONCLUSIONES	40

2.1- Introducción

En este capítulo se realizará un resumen del estado de la técnica de los dos campos en los que se enmarca esta tesis: el reconocimiento biométrico mediante huella dactilar y la aceleración hardware basada en dispositivos reconfigurables.

La huella dactilar es un identificador biométrico ampliamente utilizado y estudiado. La investigación de la comparación de huellas dactilares experimentó un auge durante el siglo pasado debido al desarrollo de sistemas automáticos para la comparación de huellas. Estos sistemas han sido profusamente utilizados tanto en el ámbito policial y forense como en el ámbito civil. La investigación en este campo ha proporcionado numerosas técnicas de comparación de huellas dactilares así como un gran abanico de sensores que proporcionan imágenes de resolución y calidad crecientes.

A lo largo de este capítulo se resumirán los conceptos básicos referentes a la comparación de huellas dactilares necesarios para la correcta comprensión de los restantes capítulos de esta tesis doctoral. Se prestará especial atención a la etapa de comparación y a las diferentes técnicas utilizadas, subdividiéndose en: técnicas basadas en minucias, técnicas basadas en crestas y técnicas de correlación. Las técnicas basadas en minucias son las más estudiadas y utilizadas. Estas técnicas realizan la comparación de las huellas dactilares mediante la identificación y localización de los puntos singulares. El resto de las técnicas de comparación han surgido como alternativa a las técnicas basadas en minucias, tratando de solventar los problemas que presentan estas técnicas, que se detallarán a lo largo de este capítulo.

De especial importancia para esta tesis son las técnicas de comparación basadas en correlación. Para estas técnicas no es necesaria una etapa de extracción de características, ya que la comparación se efectúa directamente sobre las imágenes. Esto evita la típica etapa de preprocesado requerida por las técnicas que extraen características y además posibilita la comparación de toda la información contenida

en la huella dactilar. La principal desventaja que presentan estas técnicas es su alta carga computacional.

En esta tesis además de estudiar la comparación de huellas dactilares basada en técnicas de correlación se buscarán vías para solventar su bajo rendimiento. Es por ello que en este capítulo se exploran las diferentes vías de aceleración hardware y más en concreto las posibilidades que presentan los dispositivos lógicos reconfigurables para el desarrollo de arquitecturas que solventen los problemas de rendimiento de este tipo de algoritmos.

2.2- La huella dactilar como identificador biométrico

Hoy en día las huellas dactilares se utilizan extensamente tanto en aplicaciones civiles como forenses/policiales. Esta amplia utilización y aceptación de la huella dactilar como identificador biométrico es debida en gran medida a su temprano estudio científico como medio de identificación unívoco de las personas. A finales del siglo XIX aparecieron dos estudios científicos relevantes en este ámbito: Galton, con el análisis de las huellas dactilares y la aparición de las minucias para su comparación, y Henry, con la clasificación de las huellas dactilares en función de la forma macroscópica formada por sus crestas.

A principios del siglo XX las huellas comienzan a utilizarse profusamente en la ciencia forense, facilitando a su vez a la policía la identificación de criminales. Esto conlleva la creación de bases de datos conteniendo huellas en todos los países, que experimentan un aumento considerable en el número de huellas y por tanto requieren un número creciente de expertos para su evaluación y comparación. Varios países y gobiernos ven la imperiosa necesidad de crear sistemas de reconocimiento automático de huellas dactilares (AFIS: “Automatic Fingerprint Identification System”) y empiezan las investigaciones en este ámbito a mediados del siglo XX. Este temprano conocimiento científico, unido a la amplia utilización de la huella dactilar en el ámbito forense y policial, proporcionó un importante

estudio y desarrollo de los AFIS. Este avance es palpable por ejemplo en los sensores de huella dactilar existentes en el mercado, ya que existe una variada gama de sensores con respecto a calidad de imagen, técnica de captura de la imagen e incluso precio [Biometrika]. Este avance técnico y su amplia aceptación provocaron el paso de la huella dactilar del ámbito forense/policial al ámbito de aplicaciones civiles, entre las que destaca el control de accesos.

Las huellas dactilares son la reproducción de la epidermis de la parte posterior de los dedos de la mano. Como se aprecia en la Figura 2.1, una huella dactilar está formada por un conjunto de líneas que se denominan crestas (líneas oscuras) y valles (líneas claras). Este conjunto de líneas que forman las huellas dactilares pueden asemejarse a patrones o texturas que se pueden analizar de diferentes maneras dependiendo del grado de detalle.

Las características más utilizadas para el análisis y comparación de huellas dactilares son las minucias, que son los puntos singulares que presentan las crestas. Dentro de las minucias existen diferentes tipos entre las que destacan las bifurcaciones y las terminaciones por su facilidad para la extracción automática (Figura 2.1). Las minucias se utilizaban en el reconocimiento manual que realizaban los expertos antes de existir los AFIS y es por ello que al diseñar los primeros sistemas automáticos gran parte de los algoritmos se basaron en ellas. Esta tendencia ha seguido existiendo con el paso del tiempo y las técnicas basadas en minucias son las más estudiadas y sobre las que existe una mayor cantidad de algoritmos de identificación.

Además de las minucias, las huellas dactilares presentan otros tipos de características que se pueden englobar en tres categorías, según Jain [Jain_06]:

- Nivel macroscópico o nivel 1, que son los patrones como el flujo de las crestas y el tipo de huella (arco, loop, whorl, etc). Dentro de este nivel cabe destacar el núcleo (“core”) de las huellas. Existen diversos tipos de núcleos que se suelen utilizar para clasificar las huellas y reducir con ello el tiempo de búsqueda y comparación al trabajar con grandes bases de datos.

- Nivel microscópico o nivel 2, que comprende a las minucias descritas anteriormente.
- Nivel 3: este nivel incluye todos los atributos dimensionales de las crestas excluidos en los niveles anteriores, como por ejemplo la anchura de las crestas, su forma, los poros contenidos en ellas, las crestas incipientes, los cortes, las cicatrices y otros muchos detalles permanentes presentes en las crestas. Las características de nivel 3 también son permanentes, inmutables y únicas de acuerdo con los expertos forenses y pueden aportar información relevante para la identificación.

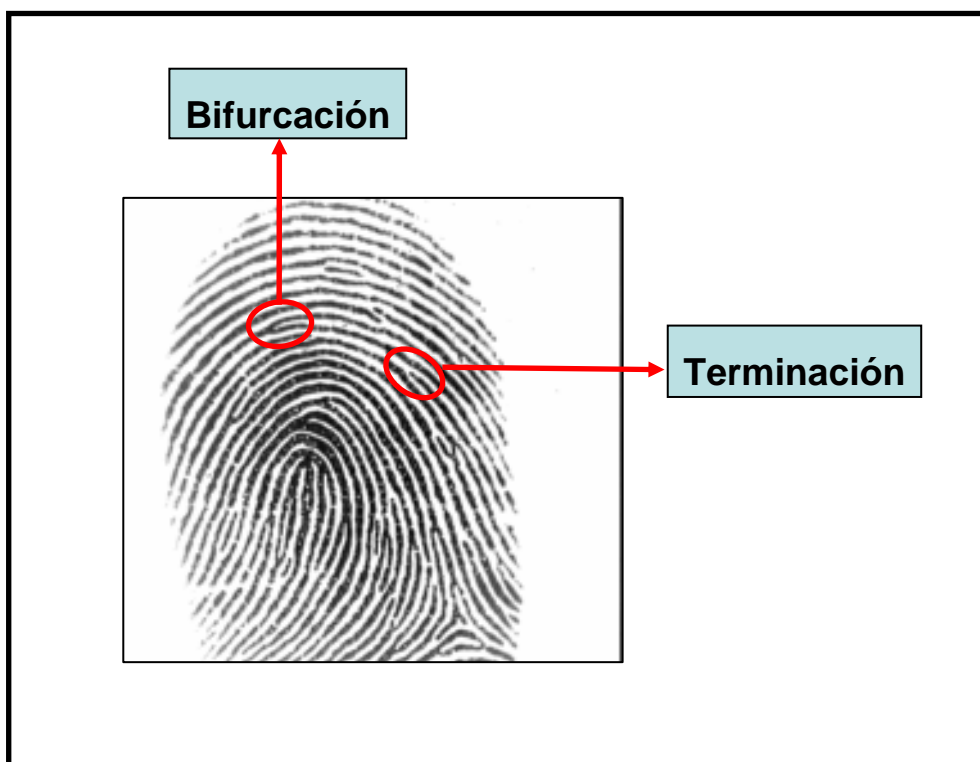


Figura 2.1. Huella dactilar.

Hoy en día la mayoría de los AFIS utilizan características de nivel 1 o 2. Esto se debe a que para la extracción de características de nivel 3 se necesitan imágenes de alta resolución que en un principio la tecnología no era capaz de proporcionar. En la actualidad existen sensores capaces de proporcionar imágenes con resolución

suficiente [Parziale_06] para la extracción de características de nivel 3. Sin embargo, los algoritmos no están preparados para su comparación de manera eficiente.

Un detallado estudio sobre las características de la huella dactilar así como el estado de la técnica en cualquier ámbito relacionado con la identificación mediante huellas dactilares se encuentra en [Maltoni_03].

2.2.1- Sistemas de reconocimiento mediante huella dactilar

En la siguiente figura se muestra un esquema típico de un sistema de comparación automática de huellas dactilares (AFIS).

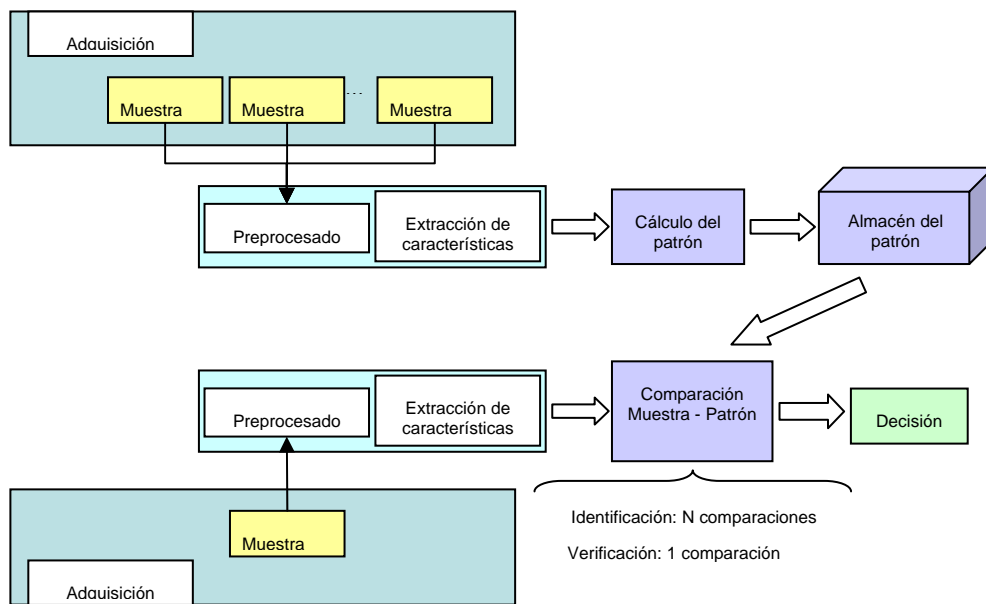


Figura 2.2. Esquema típico de AFIS

Las etapas mostradas en la Figura 2.2 se describen a continuación:

- **Adquisición:** En esta etapa se toma una muestra de la huella dactilar de un sujeto. Existen en la actualidad muy diferentes tipos de sensores (ópticos, capacitivos, térmicos, ultrasónicos, etc). Dependiendo del sensor utilizado, la calidad y la resolución de la imagen obtenida varían enormemente. Los parámetros más relevantes que se utilizan para caracterizar los sensores son la resolución, el área de captura, el número de píxeles, la precisión geométrica, el contraste y la distorsión geométrica.
- **Preprocesado:** La imagen obtenida en la etapa de adquisición se trata en función de las necesidades de los algoritmos de extracción y comparación que se vayan a utilizar. Normalmente siempre es necesario algún ajuste de la imagen para aumentar la calidad de la misma, como pueden ser ajuste de contraste, ecualización del histograma y una eliminación de las partes de la imagen que no contienen a la huella dactilar, lo que se denomina segmentación. En muchos algoritmos es necesario el cálculo del campo de orientación [Bazen_02] que determina la orientación de las crestas en cada zona de la huella dactilar. Normalmente el preprocesado requerido para los algoritmos de comparación de huellas dactilares es bastante complejo y requiere una alta capacidad computacional. Un preprocesado típico para la comparación basada en minucias se muestra en [Jain_99], [Jain_L_99]. La comparación de minucias requiere un fuerte preprocesado, ya que es necesario adelgazar las crestas hasta formar un flujo de líneas de grosor un píxel (adelgazado o thinning) para después extraer la localización de los puntos singulares.
- **Extracción de características:** En esta etapa se extrae la información relevante de las huellas para su posterior comparación. La mayor parte de los algoritmos de comparación de huellas dactilares utilizan como característica las minucias. Uno de los algoritmos de extracción de minucias más destacados es el algoritmo de seguimiento de crestas de Maio [Maio_97], que consiste en el seguimiento de todas las crestas de la huella, determinando como minucias las bifurcaciones (desdoblamiento de una

cresta en dos crestas) y terminaciones (fin de una cresta). Cabe destacar que los algoritmos de extracción de minucias extraen generalmente gran cantidad de minucias falsas que es necesario eliminar antes de pasar a la comparación para evitar errores en la decisión.

- Comparación: En esta etapa se comparan las características extraídas para determinar si las dos muestras pertenecen al mismo individuo. Existen algoritmos que no necesitan extracción de características y por tanto realizan directamente la comparación. Ya que esta parte del sistema es la que se estudia en profundidad en esta tesis, se analizarán en detalle en los siguientes apartados las diferentes técnicas de comparación de huellas dactilares.

2.2.1.2- Métricas de evaluación

Para evaluar la precisión de los algoritmos de comparación de los sistemas biométricos se suelen utilizar dos tasas de error:

- FAR (“False Acceptance Rate”) o Tasa de Falsa Aceptación, que mide la cantidad de errores que produce el algoritmo de comparación cuando reconoce muestras de distintos usuarios como pertenecientes a un mismo usuario. Esta tasa de error también se denomina FMR (“False Match Rate”).
- FRR (“False Rejection Rate”) o Tasa de Falso Rechazo, que mide los errores producidos cuando el algoritmo de comparación no reconoce muestras de un mismo usuario y las considera pertenecientes a distintos usuarios. Esta tasa de error también se denomina FNMR (“False Non Match Rate”).

En la Figura 2.3 se muestra un ejemplo de FAR y FRR. Como se aprecia en la figura, el umbral de sensibilidad elegido determina el punto de funcionamiento del

sistema y ambas tasas de error varían de forma inversa con lo que no es posible minimizar ambas al mismo tiempo. Una medida de precisión del sistema es el EER (“Equal Error Rate”), que es el punto en el que ambas tasas de error se igualan. Este punto no suele ser el de trabajo para un sistema real ya que normalmente se minimiza una de las dos tasas de error, FAR o FRR, en función de las necesidades del sistema.

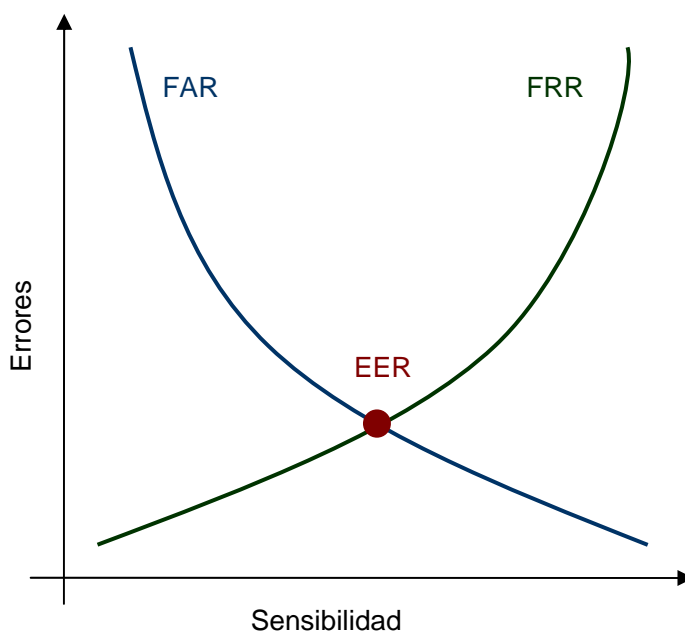


Figura 2.3 Tasas de error de un sistema biométrico.

Otra forma de representar las tasas de error es enfrentar FAR con FRR. Esta curva se denomina ROC (“Receiver Operating Characteristic curve”). Normalmente se utiliza escala logarítmica para su representación, ya que cuantos menos fallos existan en el sistema más cercana estará la curva a los ejes de coordenadas.

2.2.2- Algoritmos de comparación mediante huellas dactilares

Debido a la temprana utilización e implantación de la huella dactilar en diversos ámbitos técnicos específicos (reconocimiento forense y policial) y civiles de amplia utilización (control de acceso), existen multitud de algoritmos de comparación de huellas dactilares. Las principales técnicas de comparación de huellas dactilares se pueden dividir en tres grandes subgrupos [Maltoni_03]:

- Técnicas basadas en comparación de minucias. Son las más utilizadas y estudiadas. Básicamente consisten en conseguir el alineamiento óptimo de dos huellas para posteriormente realizar correspondencias entre el mayor número de pares de minucias.
- Técnicas basadas en comparación de características de las crestas. Estas técnicas realizan la comparación en base a otros atributos de las crestas, como pueden ser los poros, el grosor de las crestas, etc.
- Técnicas de correlación. Realizan la correlación de los píxeles de las imágenes de las huellas dactilares para determinar el grado de similitud de las imágenes.

La comparación de huellas dactilares presenta una serie de problemas comunes a todos los tipos de técnicas, ya que están asociados a las imágenes utilizadas y a las técnicas de captura. Estos problemas existentes en las muestras deben ser tratados para minimizar su efecto y se resumen a continuación:

- Desplazamiento relativo de las muestras.
- Rotación relativa de las muestras.
- Solapamiento parcial entre las muestras. Debido a los sensores de escasa superficie o a la mala captura de las muestras, pueden llegar a tomarse muestras de un mismo dedo con escaso solapamiento.

- Distorsión no lineal, debida a la elasticidad de la piel.
- Efectos en la imagen debidos a las características de la piel del dedo como pueden ser: presión, humedad, sequedad, etc.
- Ruido. Puede ser introducido por el sensor en el proceso de captura.

2.2.2.1- Técnicas basadas en minucias

Los algoritmos basados en minucias utilizan la comparación de minucias o puntos singulares de las crestas de las huellas dactilares. Son los más utilizados y estudiados, ya que son una versión automatizada del método que utilizan los expertos de la policía a nivel mundial desde hace décadas para el reconocimiento de criminales.

En este tipo de algoritmos cobra suma importancia la extracción de las minucias. En esta fase, a cada huella se le asocia un conjunto de minucias que estará formado por un vector de información para cada una de las mismas. El contenido del vector varía notablemente de un algoritmo a otro, pero es necesario que contenga la posición de la minucia acompañada por algún tipo de información relevante de la misma, como puede ser el tipo de minucia (bifurcación, terminación, etc.), orientación, parte de la cresta que lo contiene, la posición relativa respecto a otras minucias, etc. La cantidad de información contenida en el vector así como el número de minucias requeridas para cada huella determina el tamaño de la información que debe ser almacenada, lo que es un parámetro crítico para multitud de aplicaciones.

Una vez obtenidos los vectores, el algoritmo alineará los conjuntos de minucias para poder determinar el número de minucias coincidentes y con ello emitir un resultado o “score” sobre la similitud de dos muestras.

Debido a las características de la imagen de la huella dactilar, ni la extracción ni la comparación de minucias son triviales. Hay que tener en cuenta y contrarrestar

numerosos efectos indeseados descritos anteriormente, como por ejemplo: desplazamiento, rotación, distorsión no lineal, ruido, presión y estado de la piel, etc.

En realidad el problema de la comparación basada en minucias se asemeja a la comparación de patrones de puntos. En este ámbito existen diferentes técnicas entre las que destaca la comparación con prealineamiento. Ejemplo típico de este tipo de algoritmos se encuentra en [Jain_97], en el que se realiza la comparación de las minucias gracias a la alineación de los conjuntos de minucias y de una parte de las crestas que las contienen (para realizar este proceso es necesario compensar la distorsión no lineal con un algoritmo elástico adaptativo).

Existen multitud de algoritmos basados en comparación de minucias. Ejemplo de ello son: [Germain_97], algoritmos que utilizan la orientación de las minucias como [Tico_03] y [Lifeng_04], algoritmos que operan a nivel local como [Xudong_00] y [Lee_02], o algoritmos que utilizan vectores de minucias especiales [Tong_04].

Las técnicas de comparación basadas en minucias, aunque son las más utilizadas y estudiadas, presentan numerosos inconvenientes entre los que destacan:

- Carga computacional media, debida a la alta complejidad de los algoritmos necesarios para la extracción y comparación de minucias. El preprocesado necesario para este tipo de algoritmos es muy complejo y además debido a este fuerte preprocesado se pueden generar crestas y minucias falsas que perjudican a la precisión de los algoritmos.
- Generación de falsas minucias en el proceso de comparación que será necesario evaluar para su posterior descarte.
- Sesgo de la información comparada. Al fin y al cabo se está extrayendo un conjunto de puntos de una imagen para caracterizarla, despreciando el resto de la información contenida en ella.

Todos estos inconvenientes han dado lugar al estudio y desarrollo de nuevas técnicas basadas en otras características para aumentar la precisión y la eficiencia de los AFIS.

2.2.2.2- Técnicas basadas en crestas

Los algoritmos de este grupo extraen características de las crestas para realizar la comparación de las huellas. En realidad la extracción de minucias es considerada por algunos autores como un subgrupo de estos algoritmos, ya que las minucias también se extraen de ellas, pero debido a su relevancia se han considerado aparte en esta clasificación de técnicas de comparación. En un principio estos algoritmos surgieron como una alternativa a los algoritmos basados en minucias y con ellos se buscaba reducir el coste computacional ahorrando la extracción de minucias para utilizar otras características que poseyeran las crestas. Existen numerosas aproximaciones en este grupo, de entre las que destacan:

- Poros. Los poros de sudor que se localizan en las crestas cumplen los requisitos para ser un identificador biométrico, pero es necesario recurrir a sensores de alta resolución y coste para poder detectarlos [Stosz_94].
- Geometría y atributos de las crestas. Basados en la comparación geométrica de las crestas. [Xiao_86].
- Textura. Las texturas son la repetición espacial de elementos básicos y se pueden caracterizar por multitud de parámetros como son: frecuencia, escalado, orientación, simetría, etc. En una huella los cambios de textura se manifiestan de forma brusca en las regiones dónde existen discontinuidades, mientras que en el resto de la huella la frecuencia y la orientación tienen una variación suave. Estas discontinuidades en la textura manifiestan la situación de los puntos singulares de la huella.

La textura se puede analizar en el dominio de Fourier [Coetzee_93], [Willis_01]. El análisis de textura se puede hacer global o localmente, siendo la segunda opción la que proporciona mejores resultados. Para realizar un análisis local de la textura se suelen utilizar bancos de filtros como podrían ser los filtros de Gabor [Jain_00], o las wavelets [Tico_01_b], [Mokju_04], [Huang_04]. También se pueden utilizar técnicas híbridas que combinan minucias y texturas [Jain_01].

En este tipo de algoritmos se engloban gran cantidad de diferentes técnicas que se caracterizan por la necesidad de imágenes de alta resolución y una alta complejidad computacional para extraer las características. En algunos casos las características extraídas proporcionan un conjunto de información más completa que las minucias, pero su comparación y extracción no puede considerarse en la mayoría de los casos ni más sencilla ni con menor complejidad computacional.

2.2.2.3- Técnicas de correlación

Este tipo de algoritmos utiliza la correlación entre dos imágenes para medir su similitud. Básicamente, la correlación de dos imágenes acumula la comparación píxel a píxel del nivel de gris de las imágenes y puede calcularse de dos maneras que se detallan a continuación:

➤ Correlación Directa.

La correlación cruzada de dos imágenes, T e I de dimensiones n x m píxeles, se define como:

$$CC(T, I) = \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} T(i, j) I(i, j) \quad (2.1)$$

Para compensar los efectos de la presión y de la condición de la piel, en la comparación de huellas dactilares mediante técnicas de correlación se utiliza la correlación cruzada normalizada con media nula, ZNCC (“Zero-Mean Normalized Cross-Correlation”). En la siguiente fórmula se muestra la expresión que permite calcular ZNCC [Crouzil_96]:

$$ZNCC = \frac{CC(T - \bar{T}, I - \bar{I})}{\|T - \bar{T}\| \cdot \|I - \bar{I}\|} \quad (2.2)$$

en la cual \bar{T} e \bar{I} son el promedio de los píxeles de ambas imágenes.

Para evitar la distorsión no lineal provocada por la elasticidad del dedo se suele calcular la correlación localmente, ya que los efectos de la distorsión no lineal afectan a la huella globalmente pero son despreciables a nivel local. Para ello se divide la imagen en bloques y se hace la correlación de cada bloque de una imagen con todos los bloques de la otra imagen. Los bloques deben tener un tamaño suficientemente pequeño para que no les afecte la distorsión no lineal, pero tampoco deben ser excesivamente pequeños ya que entonces se obtendría alta correlación con cualquier bloque provocando resultados erróneos. El tamaño del bloque es un parámetro muy importante en estos algoritmos que debe ser determinado de acuerdo con el algoritmo y el tipo de imágenes a tratar.

Además de tener en cuenta la distorsión no lineal (que se controla con el tamaño de los bloques) es necesario tener también en cuenta la distorsión lineal (traslación y rotación) que deberá tratarse realizando la correlación de las muestras trasladadas en ambos ejes y rotadas.

[Hatano_02] utiliza comparación diferencial para aumentar el valor de la correlación: correlación del máximo-correlación del mínimo en un entorno del máximo.

Una aplicación de la correlación a nivel local se muestra en [Bazen_00]. En este trabajo se utilizan un conjunto de bloques para caracterizar el patrón y se comparan

con la imagen de entrada mediante la correlación cruzada. La comparación dará un resultado positivo si los bloques se corresponden espacialmente en ambas huellas. En cierta medida, este método se asemeja al método de las minucias, pero con la ventaja de evitar su extracción y alineamiento. Uno de los principales problemas que se presentan con este método y que es común para todos los algoritmos que utilizan la correlación como medida de similitud es la elevada carga computacional que conlleva. En este trabajo la selección de los bloques dentro del patrón se puede hacer mediante tres criterios: zonas cercanas a las minucias, zonas con baja autocorrelación (distintivas dentro de la imagen) y zonas con baja coherencia (cercanas a las minucias).

En esta línea existen otros trabajos que utilizan diferentes criterios para seleccionar bloques de la huella de manera apropiada para el cálculo de la correlación. Existen varios trabajos utilizando correlación de bloques que contienen a las minucias como [Yahagi_90], [Kovacs_00], [Huvanandana_00] y [Jiang_07].

➤ Correlación en el dominio de la frecuencia.

Para reducir la carga computacional se puede calcular la correlación cruzada en el dominio de la frecuencia. El teorema de correlación [Gonzales_92] para dos imágenes establece que:

$$CC(T, I) = F^{-1}(F^*(T) * F(I)) \quad (2.3)$$

En esta fórmula se obtiene la correlación haciendo la transformada de Fourier de ambas imágenes (T e I), una de ellas conjugada, multiplicándolas y haciendo la transformada inversa del resultado. El resultado obtenido es una matriz en la cual cada elemento representa la correlación de T e I desplazada tantos píxeles en horizontal y vertical como indique el número de fila y de columna del elemento.

De esta forma la posible traslación entre las imágenes se calcula a la vez que se realiza el cálculo de la correlación cruzada en el dominio de la frecuencia. Sería

necesario realizar el cálculo incluyendo rotaciones para considerar también este efecto muy común en las huellas dactilares.

Un ejemplo de la utilización del cálculo de la correlación en el dominio de la frecuencia para la comparación de huellas dactilares se muestra en [Seow_02], [Coetzee_93]. También hay autores que utilizan bancos de filtros de correlación para efectuar la comparación [Venkataramani_03].

Existen otros autores que realizan la correlación mediante lentes ópticas [Wilson_97], [Wilson_00], [Klima_97]. El problema de estos sistemas es el coste de los componentes además de la complejidad añadida que acarrea su ajuste y utilización.

Para evitar tener que realizar el cálculo de la correlación de diferentes rotaciones se puede utilizar la transformada de Fourier-Mellin [Chen_94], en lugar de la transformada de Fourier. Para el cálculo de esta transformada es necesario realizar la transformada de Fourier de la imagen. Sobre el módulo del resultado obtenido se realiza un cambio a coordenadas logarítmico-polares y se realiza nuevamente una transformada de Fourier obteniendo con ello la transformada Fourier-Mellin de la imagen. La transformada Fourier-Mellin es invariante para rotaciones y traslaciones, por lo tanto con un único cálculo se podría determinar CC incluyendo rotaciones y traslaciones. El problema es que esta transformada conlleva transformaciones complejas que generan una carga computacional mayor que la necesaria para realizar una transformada de Fourier [Derrode_01]. Por ejemplo, en [Zhang_06] se utiliza esta transformada para alinear las huellas. En [Ouyang_06] se utiliza la transformada Fourier-Mellin para realizar comparación de huellas a nivel local.

Los algoritmos basados en correlación están obteniendo actualmente excelentes resultados [FVC2002], [FVC2004]. Su principal desventaja es que conllevan un alto coste computacional. Esto hace que sea inviable su utilización en numerosas aplicaciones, sobre todo en aquellas con restricciones en tiempo de respuesta o velocidad de procesamiento. Motivados por el escaso estudio de estos métodos en el campo de la comparación mediante huella dactilar y sus buenos resultados

acompañados de una escasa eficiencia computacional, esta tesis se centrará en el estudio de estos métodos y de su aceleración.

2.3- Aumento de prestaciones mediante aceleración Hardware

Las técnicas basadas en correlación proporcionan muy buenos resultados en precisión para la comparación de huellas dactilares tal y como muestran los resultados de [FVC2002], [FVC2004]. El problema que presentan estas técnicas es su alta complejidad que acarrea una ineficiencia temporal, sobre todo para aplicaciones civiles en las cuales el tiempo de espera del usuario es un factor muy importante. En aplicaciones forenses la reducción del tiempo de comparación es interesante cuando se compara con una base de datos, pero en este caso el tiempo de comparación no se puede catalogar como tiempo real.

El cálculo de la correlación, fórmula 2.1, se adapta muy bien a la aceleración hardware ya que su alta complejidad computacional se puede acelerar mediante segmentación. Sin embargo no ocurre lo mismo con las técnicas basadas en minucias, que son difícilmente portables a arquitecturas segmentadas ya que necesitan un gran control de flujo. En este apartado se expondrán las diferentes vías que se pueden utilizar para acelerar algoritmos y el estado de la técnica en este ámbito relacionado con el procesado de imágenes y algoritmos de reconocimiento biométricos.

Antes de la aparición de las tecnologías reconfigurables, en el campo de la computación se podían implementar algoritmos de dos maneras:

- Vía Hardware. Gracias al diseño de circuitos a medida (ASIC, “Application Specific Integrated Circuit”) que implementan funciones específicas.

- Vía Software. Mediante la utilización del conjunto de instrucciones de un microprocesador.

La implementación de algoritmos se realiza habitualmente por software sobre un microprocesador determinado. Esta solución es la más cómoda y flexible. Sin embargo, los microprocesadores ofrecen unas prestaciones limitadas, las cuales se ponen de manifiesto cuando es preciso realizar tareas computacionalmente pesadas en tiempo real o con un tiempo de respuesta exigente. En el campo del procesamiento digital de señal, y en particular en el campo del procesamiento de imágenes, es habitual utilizar procesadores específicamente diseñados para estas tareas, conocidos como Procesadores Digitales de Señal (DSPs), que mejoran significativamente las prestaciones. A pesar de todo, los DSPs resultan insuficientes o inadecuados para muchas aplicaciones actuales. En estos casos es necesario recurrir a un soporte hardware especializado que acelere las tareas críticas.

Con los ASIC se obtienen prestaciones inmejorables, ya que en este caso se diseña un circuito para una funcionalidad específica. El problema que presentan es su elevado coste y su escasa flexibilidad, ya que cualquier cambio que se desee realizar después de fabricación implica rediseño y refabricación con el consiguiente aumento de costes asociado. En el caso de las soluciones software la flexibilidad es mayor, ya que partiendo de un conjunto de instrucciones y de un mismo hardware se puede implementar prácticamente cualquier funcionalidad. El problema en este caso es que se reducen las prestaciones (para la ejecución de cada instrucción es necesario uno o varios accesos a memoria, decodificación y ejecución de la misma) a favor de un aumento de la flexibilidad y disminución de costes (el hardware en este caso es de propósito general y puede ser utilizado para cualquier aplicación). Otro problema que presentan es que el conjunto de instrucciones se determina al fabricar el microprocesador, con lo cual, si es necesario utilizar otras instrucciones, deben ejecutarse basándose en instrucciones que ya posea el microprocesador, lo que repercute desfavorablemente en las prestaciones.

Las tecnologías reconfigurables permiten aunar las ventajas de las dos vías descritas anteriormente. La idea es obtener las ventajas en cuanto a prestaciones de los ASIC pero con la flexibilidad y disminución de costes que implica la reconfiguración o cambio de funcionalidad que podría tener un microprocesador. Las ventajas que presentan son:

- Altas prestaciones. Cercanas a las obtenidas con los ASIC y por tanto muy superiores a las que se obtienen con un microprocesador.
- Bajo coste. Debido a su capacidad de reconfiguración estos dispositivos utilizan un mismo hardware para la implementación de cualquier circuito; esto proporciona ventajas a nivel de usuario, ya que un mismo hardware es válido para múltiples diseños, y a nivel de fabricación, porque el mismo dispositivo se vende a todos los clientes reduciendo con ello los costes en ambos casos.
- Sencillez. Los dispositivos van unidos a entornos de desarrollo software que facilitan en gran medida la tarea del diseñador. En estos entornos existe la posibilidad de diseñar los circuitos mediante lenguajes de descripción Hardware como VHDL, aumentando así el nivel de abstracción del diseño.
- Flexibilidad. La flexibilidad es máxima, pudiendo implementarse en estos dispositivos prácticamente cualquier circuito digital con cualquier funcionalidad. La flexibilidad también se puede aprovechar para reconfigurar el dispositivo en tiempo de ejecución, pudiendo implementar circuitos de grandes dimensiones que no podrían ser implementados de una sola vez.
- Adaptabilidad. Debido a la capacidad de reconfiguración, estos dispositivos son ideales para aplicaciones que necesiten modificación de parámetros en tiempo de ejecución.

En resumen, las ventajas de este tipo de tecnologías son su versatilidad y flexibilidad a bajo coste, pudiendo utilizarse para implementar circuitos a medida de una manera sencilla, cómoda y barata con el consiguiente beneficio en la mejora

de prestaciones. El amplio abanico de aplicaciones en los que pueden utilizarse estos dispositivos ha dado lugar a dos grados diferentes de reconfiguración:

- Grano fino. En este caso la reconfiguración se hace a nivel de bit y por tanto la flexibilidad alcanza su máximo nivel. En estos dispositivos se puede implementar cualquier función lógica. Este tipo de reconfiguración es el que ofrecen las FPGAs (Field Programmable Gate Array).
- Grano grueso. Este tipo de reconfiguración renuncia a parte de la flexibilidad a cambio de una mayor potencia de cálculo. Estos dispositivos tienen bloques básicos con más funcionalidades que los anteriores que los hacen aptos para operaciones más complejas, ahorrando al diseñador repetir innecesariamente configuraciones a nivel de bit. Realmente es un paso intermedio entre los microprocesadores con un conjunto de instrucciones y los dispositivos reconfigurables de grano fino, con necesidad de configuración a nivel de bit.

En la actualidad empresas como Xilinx [Xilinx] y Altera [Altera] ofrecen dispositivos reconfigurables de altas prestaciones. Ejemplos de ello son las familias Virtex 4 [Xil_V4] o Virtex 5 [Xil_V5] de Xilinx. Ambas familias se subdividen en otras tres familias: LX (lógica de altas prestaciones), SX (procesado de señal de altas prestaciones) y FX (procesamiento empotrado). De entre las características generales de la familia Virtex 5 [Xil_V5] destacan:

- El número de puertas lógicas, siendo en la gama más alta de hasta 330.000 celdas lógicas, que contienen: hasta 207.360 flip-flops y hasta 207.360 LUTs (“Look-Up table”) de 6 entradas.
- Poseen bloques de memoria embebidos de gran capacidad (36 kb), pudiendo llegar a 4.200 kb en el dispositivo mayor de la gama más alta.

- Altas prestaciones para procesado digital de la señal (DSP), gracias a los Xtreme DSP slices que tienen capacidad para configurar hasta 40 operaciones de DSP y aritméticas (grano grueso).
- Procesamiento mediante microprocesador empotrado.
- Altas prestaciones para configuraciones rápidas y eficientes en puertos de entrada/salida así como conectividad entre bloques.
- Frecuencia de reloj de hasta 550 MHz.

A lo largo de los últimos años se ha demostrado que las tecnologías reconfigurables proporcionan una excelente mejora de prestaciones a la hora de implementar ciertos algoritmos de diversos campos específicos como por ejemplo en el de la encriptación de datos [Elbirt_00], o en el de la compresión de datos [Huang_00], tecnologías inalámbricas, procesado digital de señal, sobre todo procesado de imágenes [Memik_03], [Vancourt_05], [Diaz_06], [Gupta_06], [Choi_06], [Saegusa_08], [Bowen_08], reconocimiento de patrones [Liang_03], [Bariamis_04], biometría [Nguyen_06], procesado de imágenes médicas [Dillinger_06], implementación de transformadas matemáticas [Uzun_05], [Jiang_M_07] y un amplio etcétera.

En el campo del reconocimiento biométrico mediante huella dactilar, existen varios trabajos que describen implementaciones en FPGAs de algoritmos de comparación y preprocesado, o de etapas de los mismos. Destaca el trabajo de Ratha y A.K. Jain [Ratha_96], por ser un trabajo pionero en este campo. En este caso se realiza una implementación del sistema en una FPGA para acelerar la comparación de huellas dactilares de bases de datos de gran tamaño.

Buscando una mejora en prestaciones, cabe notar que las tecnologías reconfigurables abren paso a nuevas arquitecturas que pueden dar lugar a híbridos entre microprocesadores, ASIC y dispositivos reconfigurables que dependen fuertemente de la funcionalidad específica que se quiera implementar. Ejemplo de estos sistemas implementando algoritmos o parte de algoritmos para comparación de huellas dactilares se muestran en [Yoo_07], [Barrenechea_07], [Fons_07],

[Lopez_06]. Estos sistemas utilizan microprocesadores, empotrados o no en la FPGA, que ejecutan el código de comparación de huellas dactilares y para las partes computacionalmente complejas utilizan hardware a medida ubicado en la FPGA. Este modelo de sistema es común a muchas arquitecturas y lo que busca es la aceleración hardware además de un interfaz eficiente en las comunicaciones.

En el campo de la aceleración hardware de algoritmos o de parte de los mismos, muchos investigadores han centrado sus esfuerzos en la reconfiguración del sistema en la medida que el algoritmo lo necesite, es decir en la reconfiguración dinámica [Brunelli_08]. En este ámbito, los desarrollos se pueden enfocar en las dos vertientes comentadas anteriormente: reconfiguración de grano fino y reconfiguración de grano grueso.

La reconfiguración de grano fino se caracteriza por la necesidad de descargar el bitstream cada vez que se necesite reconfigurar, incrementando considerablemente el tiempo de ejecución. Muchos autores, para evitar este inconveniente, subdividen la FPGA en zonas y realizan reconfiguraciones parciales. En realidad, en todos estos sistemas para conseguir una aceleración es necesario que el tiempo de reconfiguración sea mucho menor que el tiempo de computación requerido por el algoritmo. En caso contrario la reconfiguración del dispositivo afecta negativamente a la eficiencia temporal del algoritmo implementado en una FPGA.

Ejemplos de la utilización de mecanismos de reconfiguración de grano fino se muestran en [Tan_08], [Jean_99], [Sedcole_06]. Al igual que en otro tipo de sistemas diseñados con FPGAs existen multitud de campos de aplicación, como por ejemplo las comunicaciones [McDonald_08], la automoción [Becker_07] y un largo etcétera.

La reconfiguración de grano grueso se caracteriza por no reconfigurar a nivel de bit la FPGA. En este caso se realizan bloques adaptables a los requerimientos del sistema pero con una menor flexibilidad. Con esta aproximación se intenta reducir el tiempo de reconfiguración a cambio de un recorte en las posibilidades de reconfiguración.

Ejemplos de la utilización de mecanismos de este tipo de reconfiguración dinámica se muestran en [Kitaoka_03], [Hartenstein_00].

Partiendo de las premisas expuestas en apartados anteriores y a la vista de las prestaciones que ofrecen actualmente las tecnologías reconfigurables se puede concluir que las FPGAs presentan numerosas ventajas para la implementación de algoritmos de reconocimiento mediante huella dactilar. De entre estas ventajas cabe destacar las siguientes:

- Capacidad de procesamiento digital de la señal de muy altas prestaciones (varios ordenes de magnitud superiores a microprocesadores o DSPs) mediante ejecución paralelo o segmentada.
- Posibilidad de empotrarse dentro de un ASIC (para aplicaciones de tirada masiva y bajo coste) o en tarjetas aceleradoras.
- Posibilidad de SoPC (“System on Programmable Chip”), es decir un sistema con un microprocesador empotrado dentro de una FPGA.
- Capacidad intrínseca de reconfiguración aprovechable para la implementación de algoritmos o partes de los mismos, pudiendo rediseñar el circuito en tiempo real a medida que la ejecución del algoritmo lo necesite.

2.4- Conclusiones

Actualmente la huella dactilar es un identificador biométrico que se utiliza extensamente tanto en aplicaciones forenses como civiles. Su gran utilización es debida a que es el primer identificador biométrico reconocido en el ámbito forense y por tanto ha sido más estudiado que otros identificadores de aparición más reciente. A medida que el progreso de los computadores y las TIC lo permitía, se han ido desarrollando sistemas de reconocimiento automático de huellas dactilares (AFIS).

Gran parte de los métodos de reconocimiento para la huella dactilar se basan en el reconocimiento de minucias (singularidades dentro del flujo de las crestas de una huella), debido a que el reconocimiento por minucias era la base del reconocimiento manual realizado por expertos. Para los métodos basados en minucias es necesario realizar unas etapas de preprocesado y extracción de características de gran complejidad computacional. Además de esto, al extraer las minucias de la imagen de la huella dactilar proporcionada por el sensor se está limitando la información contenida en la imagen a un conjunto de puntos singulares, apartando con ello el resto de la información que contiene la imagen.

Para paliar estos efectos surgen las técnicas basadas en la correlación de imágenes para la comparación de huellas dactilares. Con estas técnicas no se realiza una etapa de extracción de características propiamente dicha, ya que se mide la similitud entre dos imágenes directamente con la imagen o con partes de la misma. Con estas técnicas es posible comparar el contenido de la imagen en su totalidad y no limitándose a un conjunto específico de características extraídas. Las tasas de error proporcionadas con estas técnicas son bastante buenas (el ganador del FVC 2004 [FVC2004] utiliza un algoritmo basado en técnicas de correlación). El mayor problema que presentan estas técnicas es su elevado coste computacional, ya que se está midiendo la similitud de los píxeles de la imagen y por tanto la complejidad computacional es directamente proporcional al tamaño de las imágenes a comparar.

Hoy en día en el campo de la huella dactilar existen sensores con muy alta resolución capaces de proporcionar imágenes de alta calidad e incluso existen sensores sin contacto [Parziale_06] que eliminan efectos indeseables que aparecen típicamente en las imágenes capturadas con sensores por contacto (como por ejemplo la distorsión no lineal, etc). Estas imágenes de alta resolución proporcionan una mayor cantidad de información que la que es posible comparar con los algoritmos de comparación basados en minucias. Es por tanto necesario adaptar los algoritmos de comparación de huellas dactilares a la comparación de la totalidad de la información contenida en la huella. En esta tesis se ha optado por las técnicas de correlación para realizar esta adaptación, debido fundamentalmente a tres factores:

- Desde el punto de vista investigador es un campo más novedoso que el de los algoritmos basados en minucias, ya que este tipo de algoritmos son menos utilizados y han sido menos estudiados por su aparición más tardía en este campo. Los algoritmos de correlación están asociados al campo de tratamiento de imágenes, pero no tanto en su vertiente biométrica. Ya que las imágenes biométricas presentan unas características especiales frente al resto de las imágenes, el estudio de las técnicas de correlación para este tipo de imágenes reviste un especial interés.
- Los algoritmos de correlación comparan la imagen en su totalidad, por tanto comparan toda la información contenida en ella. Esto aporta a la comparación biométrica tanto factores positivos (un aumento en la seguridad del sistema ya que se compara toda la información y no un subconjunto de la misma) como negativos (debido a la complejidad de las imágenes de huellas dactilares y los efectos producidos en ellas la comparación con este tipo de técnicas acarrea problemas que deberán ser estudiados y tratados para ser eliminados en la medida de lo posible).
- Uno de los problemas que presentan las técnicas de correlación es su elevado coste computacional. Por tanto, un aspecto crucial es el aumento de prestaciones, tanto a nivel de algoritmos como a nivel de aceleración hardware mediante sistemas específicos. El coste computacional lleva asociado un aumento en el tiempo de ejecución que necesita ser paliado con equipos de gama alta. A su vez esto implica un alto coste en el desarrollo de un sistema comercial.

Por todas estas razones esta tesis se centra en el estudio de algoritmos de correlación para la comparación de huellas dactilares y su aceleración hardware. Se comenzará por el estudio y diseño de algoritmos de comparación de huellas dactilares basados en técnicas de correlación para aportar soluciones a los problemas que se han presentado anteriormente. A partir de los algoritmos se estudiarán y diseñarán tanto arquitecturas hardware como sistemas empujados para su aceleración.

Capítulo 3:

COMPARACIÓN DE HUELLAS DACTILARES MEDIANTE TÉCNICAS DE CORRELACIÓN

3.1-	INTRODUCCIÓN.....	46
3.2-	ALGORITMO DE COMPARACIÓN PARA IMÁGENES DE BAJA RESOLUCIÓN	49
3.2.1-	<i>Preprocesado</i>	53
3.2.2-	<i>Alineamiento</i>	56
3.2.3-	<i>Selección de zonas</i>	58
3.2.4-	<i>Comparación</i>	60
3.3-	ALGORITMO DE COMPARACIÓN PARA IMÁGENES DE ALTA RESOLUCIÓN.....	62
3.3.1-	<i>Selección de la información dentro de la huella</i>	63
3.3.2-	<i>Selección de zonas</i>	70
3.3.3-	<i>Comparación</i>	76
3.4-	RESULTADOS EXPERIMENTALES	78
3.4.1-	<i>Algoritmo de comparación para huellas de baja resolución</i>	78
3.4.2-	<i>Algoritmo de comparación para huellas de alta resolución</i>	81
3.5-	CONCLUSIONES	88

3.1- Introducción

Las técnicas más utilizadas para comparar huellas dactilares se basan en la extracción y localización de minucias. Estas técnicas, que imitan a la comparación de huellas realizada manualmente, se basan en determinar los puntos singulares de las crestas de las huellas y comparar su localización. Las etapas de extracción y comparación de minucias son etapas complejas, ya que es necesario tener en cuenta, entre otros muchos efectos, la distorsión no lineal que provoca la elasticidad de la piel, el ruido introducido por el sensor, el estado de la piel, etc. Por otra parte, no es trivial determinar un punto de referencia en ambas huellas, lo cual es necesario para alinear las huellas, que pueden estar rotadas y desplazadas entre sí.

Sin embargo, las técnicas de comparación basadas en correlación no requieren extracción de características de las huellas dactilares a comparar. Estas técnicas utilizan el cálculo de la correlación cruzada para determinar la similitud entre dos imágenes. La principal desventaja que presentan frente a otros métodos es la elevada carga computacional que conlleva el cálculo de la correlación cruzada.

Esta tesis se ha orientado a la utilización de técnicas basadas en correlación de imágenes frente a las basadas en minucias, que son técnicas más maduras y sobre las que existen numerosos trabajos de investigación. Esta decisión se ha apoyado en diversos factores, entre los que destaca el mejor rendimiento que presentan las técnicas de correlación en plataformas hardware en comparación con las técnicas basadas en minucias [Entrena_04], [Lindoso_06], [Lindoso_07_a]. Esto es debido a que el control de flujo en este caso es mucho menor que el necesario para las técnicas basadas en minucias, que en definitiva se basan en la extracción, localización y comparación de una característica extraída de una huella. Otro factor importante a tener en consideración son los buenos resultados de precisión obtenidos por algoritmos basados en técnicas de correlación en los últimos FVC [FVC2002], [FVC2004].

Aparte de los factores basados en capacidad de aceleración y buenos resultados de precisión, existe un factor si cabe aún más importante. Utilizando las técnicas de correlación se reduce el preprocesado al que se somete la imagen. Esto implica dos consecuencias importantes: la imagen a comparar no incluye artefactos derivados del preprocesado y se reduce la carga computacional de esta etapa. Esto lleva a un planteamiento desde el punto de vista teórico diametralmente opuesto a las técnicas basadas en minucias.

En las técnicas basadas en minucias es necesario un fuerte preprocesado para modelar una huella real a imagen y semejanza de una “huella ideal”, pasando por numerosos filtrados que hacen aparecer artefactos y que en muchos casos generan crestas y minucias falsas incrementando con ello los errores. Con las técnicas basadas en correlación no se busca el filtrado respecto a una “huella ideal”, se elimina ese filtrado artificial para trasladar esa carga computacional a la etapa de comparación. En realidad el cálculo de la correlación se asemeja a un filtrado, pero sustituyendo los coeficientes de los filtros ideales por la imagen con la que se quiere comparar. Es por esto que se puede considerar como un “filtrado real” entre las dos huellas que se van a comparar. Este cambio conceptual además aporta una riqueza a la comparación en sí misma, ya que no se busca la correspondencia de una serie de puntos extraídos a partir de una huella modificada mediante el preprocesado, sino que se busca una correspondencia global de toda la información contenida en la huella original sin necesidad de preprocesar. Debido a esto, las técnicas basadas en correlación se presentan como candidatas a la exploración de imágenes de alta resolución en el campo de la comparación de huellas dactilares. Aplicando estas técnicas es posible comparar todas las características presentes en las imágenes sin necesidad de extraerlas ni compararlas una a una como hacen la mayoría de los algoritmos en este campo.

En este capítulo se presentan dos soluciones originales para la comparación de huellas dactilares mediante técnicas de correlación. Los dos algoritmos propuestos han sido diseñados en función de la resolución de las imágenes utilizadas, proponiendo soluciones diferenciadas para imágenes de baja y alta resolución. En ambas soluciones se han establecido dos requisitos de diseño que favorezcan alta

eficiencia en una plataforma hardware: reducir en la medida de lo posible la carga computacional debida al cálculo de la correlación y utilizar el cálculo de la correlación en el mayor número posible de etapas.

El algoritmo propuesto para huellas de baja resolución se compone de cuatro etapas: preprocesado, alineamiento, selección de zonas y comparación. Debido a la baja resolución de las imágenes es necesario realizar un preprocesado ligero para adecuar la calidad de la imagen a la requerida por las etapas posteriores. La etapa de alineamiento se realiza mediante un método novedoso basado en la correlación del campo de orientación de las huellas. Después de esto, se realiza la selección de zonas, que determina las zonas de la imagen aptas para ser comparadas. Una vez seleccionadas las zonas, se realiza la comparación mediante el cálculo de ZNCC. En este algoritmo los dos requisitos de diseño se han satisfecho, ya que se utiliza el cálculo de la correlación cruzada para las etapas de comparación y alineamiento, y gracias a las etapas de alineamiento y selección de zonas se reduce considerablemente la carga computacional. Además, en la etapa de alineamiento la correlación cruzada se calcula sobre el campo de orientación que tiene dimensiones menores que la imagen, reduciendo aún más la carga computacional.

El algoritmo propuesto para huellas de alta resolución se compone de dos etapas: selección de zonas y comparación. En este caso, se selecciona una zona que contenga al núcleo de la huella. Esta selección se realiza mediante un método novedoso que utiliza la transformada wavelet sobre la imagen sin preprocesar. En este algoritmo también se realiza la comparación de las zonas seleccionadas mediante el cálculo de ZNCC. En este caso, las operaciones realizadas en cada etapa difieren, aunque cabe reseñar la similitud de las operaciones transformada wavelet (filtrado bidimensional) y correlación cruzada.

En los siguientes apartados de este capítulo se detallan los algoritmos propuestos para huellas de baja resolución apartado 3.2, y para huellas de alta resolución, apartado 3.3, así como los resultados experimentales conseguidos para ambos algoritmos, apartado 3.4.

3.2- Introducción a la correlación de imágenes

La comparación mediante técnicas de correlación utiliza la correlación cruzada, CC, para determinar la similitud de imágenes [Gonzales_92]. El cálculo de la correlación cruzada del patrón o *template*, T, con la imagen de entrada o *input*, I se calcula mediante la ecuación 3.1.

$$CC(T, I) = \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} T(i, j) I(i, j) \quad (3.1)$$

siendo ambas imágenes, T e I, de tamaño n x m.

A pesar de que la correlación cruzada es un cálculo que proporciona buenos resultados a la hora de comparar imágenes, si se consideran huellas dactilares es necesario tener en cuenta factores adicionales. A las imágenes de las huellas dactilares proporcionadas por los sensores les afectan las variaciones en la iluminación, la elasticidad de la piel, la distorsión, la presión y otros muchos factores. Para evitar que se aprecien los efectos de estas variaciones, en la comparación se suele utilizar la correlación cruzada normalizada con media nula, “Zero Mean Normalized Cross-Correlation”, ZNCC [Crouzil_96], [Maltoni_03]. ZNCC se define de la siguiente manera:

$$ZNCC = \frac{CC(T - \bar{T}, I - \bar{I})}{\|T - \bar{T}\| \cdot \|I - \bar{I}\|} \quad (3.2)$$

En la ecuación 3.2:

- CC Es la correlación cruzada.
- \bar{T} Es la media del patrón.

- \bar{I} Es la media de la imagen de entrada.

Para comparar dos huellas dactilares correctamente es necesario tener en cuenta posibles desplazamientos y rotaciones relativos entre las huellas. El cálculo de ZNCC incluyendo desplazamiento y rotación se muestra en la ecuación 3.3:

$$ZNCC(x, y, \alpha) = \frac{CC(T - \bar{T}, I(x, y, \alpha) - \bar{I}(x, y, \alpha))}{\|T - \bar{T}\| \cdot \|I(x, y, \alpha) - \bar{I}(x, y, \alpha)\|} \quad (3.3)$$

En la ecuación 3.3:

- CC es la correlación cruzada.
- \bar{T} Es la media del patrón.
- $I(x, y, \alpha)$ Es la imagen de entrada desplazada horizontalmente x píxeles y verticalmente y píxeles, y rotada un ángulo α .

La ecuación 3.3 se aplica a las imágenes desplazadas y rotadas obteniendo un conjunto de resultados para ZNCC. El valor máximo de ese conjunto de resultados establece el grado de similitud entre ambas imágenes. Los valores de (x, y, α) para los cuales se obtiene ZNCC máximo determinan el desplazamiento (x, y) y la rotación (α) entre ambas imágenes.

Otros efectos que se manifiestan en las imágenes de las huellas dactilares, como por ejemplo la distorsión no lineal global, pueden evitarse realizando el cálculo de ZNCC de manera local. Para ello se subdivide una de las imágenes y se aplica la ecuación 3.3 a cada una de las partes.

Como alternativa, el cálculo de la correlación cruzada puede realizarse en el dominio de la frecuencia tal y como muestra la ecuación 3.4:

$$CC(T, I) = F^{-1}(F^*(T) * F(I)) \quad (3.4)$$

En la ecuación 3.4:

- $F^*(T)$: Es el complejo conjugado de la transformada de Fourier del patrón.
- $F(I)$: Es la transformada de Fourier de la imagen de entrada.
- $CC(T, I)$: Es una matriz de resultados de CC. La posición dentro de la matriz indica el desplazamiento que se ha aplicado al patrón para obtener el resultado.

Los resultados de las ecuaciones 3.3 y 3.4 son equivalentes pero suele ser más eficiente realizar el cálculo en el dominio de la frecuencia ya que los desplazamientos están incluidos en el cálculo de CC tal y como muestra la ecuación 3.4.

La problemática asociada al cálculo de la correlación de imágenes está directamente relacionada con el gran número de operaciones que es necesario realizar. El número de cálculos por estimación tiene una complejidad de $O(N^2 \cdot R)$, siendo N^2 el tamaño de la imagen y R el número de rotaciones. Por lo tanto es necesario reducir en la medida de lo posible los cálculos a realizar, sobre todo mediante una etapa de alineamiento adecuada.

3.3- Algoritmo de comparación para imágenes de baja resolución

El algoritmo de comparación propuesto para imágenes de baja resolución se compone de cuatro etapas: preprocesado, alineamiento, selección de zonas y comparación [Lindoso_07_b].

Cada etapa del algoritmo realiza las siguientes funciones:

- Preprocesado: adecua la calidad de la imagen a las etapas posteriores. Como subproducto se obtiene el campo de orientación que se utiliza en etapas posteriores.
- Alineamiento: determina desplazamiento y rotación entre ambas huellas. Se realiza mediante el cálculo de la correlación cruzada de los campos de orientación de las huellas.
- Selección de zonas: determina zonas de la imagen aptas para ser comparadas. Las zonas óptimas son las que contienen mayor cantidad de información, son distintivas, y tienen buena calidad de imagen. Se realiza mediante la multiplicación de la coherencia de la orientación de ambas huellas teniendo en cuenta desplazamiento y rotación determinados en la etapa de alineamiento.
- Comparación: realiza la comparación mediante el cálculo de la correlación cruzada de las zonas seleccionadas en la etapa previa.

El diagrama de bloques del algoritmo de comparación propuesto se muestra en la Figura 3.1.



Figura 3.1. Diagrama de bloques del algoritmo de comparación

3.2.1- Preprocesado

Los algoritmos de comparación basados en técnicas de correlación pueden utilizar las imágenes directamente sin realizar ningún paso previo de extracción de características. La extracción de características es un proceso costoso a nivel computacional y que conlleva asociado un fuerte preprocesado. Por ejemplo para la extracción de minucias un preprocesado típico requiere las siguientes etapas: segmentación, binarización, adelgazamiento, cálculo del campo de orientación, etc [Jain_99].

Aunque en este caso se evite la extracción de características, es necesario garantizar que la calidad de la imagen es lo suficientemente buena para realizar el cálculo de la correlación. Esto se consigue gracias a un preprocesado ligero.

El preprocesado que se ha utilizado está basado en [Hong_98], cuyas etapas principales son: normalización, filtrado de baja frecuencia, estimación del campo de orientación y frecuencias de las crestas con sus respectivas coherencias, filtrado de Gabor y ecualización. Este proceso se muestra en la Figura 3.2.



Figura 3.2. Huella preprocesada

Cabe destacar dentro de este proceso la estimación del campo de orientación, ya que será utilizada por etapas posteriores del algoritmo de comparación. El campo de orientación suministra información local sobre la frecuencia y orientación de las crestas de la huella. Por la importancia para la discusión posterior a continuación se resumirá el proceso de cálculo del campo de orientación, basado en [Jain_97].

Para el cálculo del campo de orientación es necesario dividir la imagen de la huella dactilar en bloques de tamaño $W \times W$. Se calculan los gradientes en las direcciones horizontal, G_x , y vertical, G_y , en cada píxel. A partir de los gradientes se estima la orientación, θ , en cada píxel con las siguientes ecuaciones:

$$V_X(i, j) = \sum_{u=i-\frac{W}{2}}^{i+\frac{W}{2}} \sum_{v=j-\frac{W}{2}}^{j+\frac{W}{2}} 2G_X(u, v)G_Y(u, v) \quad (3.5)$$

$$V_Y(i, j) = \sum_{u=i-\frac{W}{2}}^{i+\frac{W}{2}} \sum_{v=j-\frac{W}{2}}^{j+\frac{W}{2}} (G_X^2(u, v) - G_Y^2(u, v)) \quad (3.6)$$

$$\vartheta(i, j) = \frac{1}{2} \tan^{-1} \left(\frac{V_X(i, j)}{V_Y(i, j)} \right) \quad (3.7)$$

Se puede calcular la coherencia o el nivel de consistencia de la orientación obtenida gracias a la fórmula 3.7, tal y como muestra la siguiente ecuación:

$$C(i, j) = \frac{1}{N} \sqrt{\sum_{(i', j') \in D} |\vartheta(i', j') - \vartheta(i, j)|^2} \quad (3.8)$$

En la fórmula 3.8, D son los bloques de alrededor del bloque en el cual se está midiendo la coherencia y N es el número de bloques que contiene D.

En la Figura 3.3 se muestra el campo de orientación de una huella.



Figura 3.3. Huella de entrada y su campo de orientación

3.2.2- Alineamiento

El propósito de la etapa de alineamiento es determinar una estimación preliminar de desplazamiento y rotación entre el patrón y la imagen de entrada. El alineamiento reduce de manera drástica el número de correlaciones que se deben realizar en la etapa de comparación. La estimación de desplazamiento y rotación realizada en esta etapa no necesita excesiva precisión ya que el cálculo se puede refinar en la etapa de comparación.

La mayoría de los métodos de alineamiento de huellas dactilares utilizan características extraídas de las imágenes [Maltoni_03], [Park_03]. Sin embargo este algoritmo no se basa en la comparación de características y por tanto no tiene sentido su extracción solo para realizar el alineamiento de las imágenes. Por lo tanto el alineamiento debe basarse en información primaria de la huella. En particular, el campo de orientación puede servir para este propósito.

El campo de orientación muestra la orientación de las crestas de la huella a nivel local. La base del método de alineamiento que se propone es utilizar esa información para estimar de forma global en qué lugar de la imagen se encuentran las huellas a comparar. Para ello se calcula la correlación cruzada de las orientaciones de ambas huellas considerando desplazamiento y rotación. Para evitar discontinuidades se utiliza el ángulo doble.

El cálculo realizado se muestra en la ecuación 3.9:

$$(\Delta x, \Delta y, \Delta \alpha) = \max_{x, y, \alpha} \left(\frac{CC_{sen}(x, y, \alpha) + CC_{cos}(x, y, \alpha)}{CC_{coh}(x, y, \alpha)} \right) \quad (3.9)$$

El numerador de la ecuación 3.9 se determina gracias a las siguientes ecuaciones:

$$CC_{sen}(x, y, \alpha) = CC(\sin_W(2\theta_T), \sin_W(2\theta_I^{x,y,\alpha})) \quad (3.10)$$

$$\sin_W(2\theta_I^{x,y,\alpha}) = \sin(2\theta_I^{x,y,\alpha}) * Coh(\theta_I^{x,y,\alpha}) \quad (3.11)$$

$$\sin_W(2\theta_T) = \sin(2\theta_T) * Coh(\theta_T) \quad (3.12)$$

$$CC_{cos}(x, y, \alpha) = CC(\cos_W(2\theta_T), \cos_W(2\theta_I^{x,y,\alpha})) \quad (3.13)$$

$$\cos_W(2\theta_I^{x,y,\alpha}) = \cos(2\theta_I^{x,y,\alpha}) * Coh(\theta_I^{x,y,\alpha}) \quad (3.14)$$

$$\cos_W(2\theta_T) = \cos(2\theta_T) * Coh(\theta_T) \quad (3.15)$$

La notación utilizada en las ecuaciones 3.10 – 3.15 es la siguiente:

- $\theta_I^{x,y,\alpha}$: es la estimación de la orientación de la huella de entrada, desplazada (x, y) píxeles y rotada un ángulo α .
- θ_T : es la estimación de la orientación de la huella patrón.
- $Coh(\theta_T)$: es la coherencia de la orientación de la huella patrón.

El denominador de la ecuación 3.9, $CC_{coh}(x, y, \alpha)$, es la correlación cruzada de la coherencia de las orientaciones de ambas huellas considerando desplazamiento y rotación.

En la ecuación 3.9, el máximo de correlación determina el mejor desplazamiento (Δx , Δy) y rotación ($\Delta \alpha$) entre ambas huellas. Para evitar discontinuidades se utiliza la CC de los senos y cosenos del ángulo doble y además se pondera por la CC de las coherencias para dar mayor peso a las zonas con mayor calidad de imagen (zonas con coherencia más alta).

Una explicación intuitiva de la ecuación 3.9 se puede realizar para el caso en que el campo de orientación se redujera a un único valor, es decir, la orientación promedio de toda la huella. Considerando además que las coherencias tuvieran valor unidad, se obtendría:

$$\begin{aligned} CC_{sen}(x,y,\alpha) + CC_{cos}(x,y,\alpha) &= \sin(2\theta_T) \cdot \sin(2\theta_I^{x,y,\alpha}) + \cos(2\theta_T) \cdot \cos(2\theta_I^{x,y,\alpha}) \\ &= \cos(2\theta_T - 2\theta_I^{x,y,\alpha}) \end{aligned}$$

El máximo para este caso se obtendría obviamente para $2\theta_T = 2\theta_I^{x,y,\alpha}$, es decir, cuando se alinean las huellas. En el caso general la fórmula 3.9 calcula una media de la rotación relativa para cada zona de las huellas, la cual está ponderada por la coherencia o grado de confianza que tiene la orientación en cada zona.

La carga computacional generada con este cálculo es mucho menor que la que requiere la correlación de la imagen completa, ya que el campo de orientación tiene dimensiones mucho menores que la imagen. Las dimensiones dependen del tamaño de la ventana utilizada para el cálculo del campo de orientación que generalmente se determina en función de las características de las imágenes utilizadas. Gracias a la etapa de alineamiento se reduce considerablemente la carga computacional en la etapa de comparación.

De forma alternativa, el alineamiento se podría realizar considerando la frecuencia estimada de las crestas en vez de la orientación. Sin embargo los resultados que se obtienen son menos precisos ya que la estimación de la frecuencia se ve más afectada por la presencia de ruido que la estimación de la orientación.

Este método de alineamiento es una aportación original de esta tesis.

3.2.3- Selección de zonas

En esta etapa ambas huellas se analizan para determinar regiones candidatas para la etapa de comparación. El objetivo de esta etapa es evitar el cálculo de la correlación cruzada sobre toda la imagen, que conllevaría un alto coste computacional, y también evitar zonas de poca calidad que puedan dar lugar a errores en la etapa de comparación.

Las zonas elegidas deben cumplir dos requisitos: ser distintivas y tener buena calidad. En [Bazen_00] se presentan y comparan diferentes métodos de selección de zonas. Una forma sencilla de realizar la búsqueda de zonas distintivas es realizando el cálculo de la autocorrelación de ambas huellas. El inconveniente más destacable

de este método es el alto coste computacional que conlleva. También se pueden seleccionar zonas alrededor del núcleo (core) o zonas en las cuales las crestas tengan una gran curvatura. En este caso los resultados pueden verse afectados por el ruido en gran medida, perjudicando con ello la decisión final en la etapa de comparación.

La selección de zonas que se realiza en el algoritmo propuesto se basa en la calidad de la imagen y en el solapamiento de las huellas. La calidad de las imágenes es un factor muy importante al realizar la búsqueda de zonas candidatas para la comparación mediante correlación cruzada. Si la zona elegida se corresponde con un área de baja calidad en alguna de las dos huellas el proceso de verificación se ve afectado negativamente. Para tener en cuenta la calidad de la imagen se ha utilizado la coherencia del campo de orientación como medida de calidad de las huellas [Lim_04]. Concretamente, la operación realizada es la multiplicación de la coherencia de la orientación de ambas huellas. Este cálculo se realiza únicamente para las zonas que se solapan de ambas huellas. El solape se determina gracias al desplazamiento y rotación determinados en la etapa de alineamiento.

Un segundo aspecto fundamental es conseguir que las zonas seleccionadas sean distintivas. Cabe destacar que las zonas de más alta coherencia pueden ser poco distintivas (por ejemplo, un conjunto de crestas paralelas sin contener ningún punto singular). La solución adoptada consiste en realizar un filtrado promedio sobre los mapas de coherencia. De esta forma se permite la selección de zonas de baja coherencia siempre y cuando sus vecinos tengan alta coherencia. Por ejemplo, una zona puede tener buena calidad y baja coherencia porque existe una fuerte curvatura o muchos puntos singulares. El filtrado promedio de la coherencia dará un resultado alto para esta zona si sus zonas vecinas son de buena calidad. En cambio, si sus zonas vecinas también son de baja calidad, concluiremos que la zona no es fiable y que sus elementos distintivos son probablemente espúreos.

Cabe destacar que si dos muestras provenientes del mismo dedo están excesivamente rotadas o desplazadas, esto puede dar lugar a un solapamiento mínimo o incluso nulo, ya que las imágenes pueden mostrar zonas no coincidentes del mismo dedo. Este efecto suele venir acompañado de una calidad insuficiente de

las zonas solapadas. Con estas condiciones la comparación se realizará sobre las zonas que solapen pero es muy probable que la etapa de comparación no consiga reconocer la coincidencia entre dos huellas que pertenecen al mismo dedo.

3.2.4- Comparación

Esta es la etapa final en la cual se realiza el cálculo de la correlación cruzada sobre los píxeles de la huellas de entrada y patrón. El cálculo se realizará únicamente en las zonas seleccionadas por la etapa anterior.

Denominaremos a estas zonas área de búsqueda en la huella patrón y regiones candidatas en la huella de entrada. El área de búsqueda tiene un tamaño mayor que las regiones candidatas, para poder comparar considerando rotación y traslación de las regiones candidatas dentro del área de búsqueda. La Figura 3.4 muestra el área de búsqueda y una zona candidata para dos muestras de la misma huella.

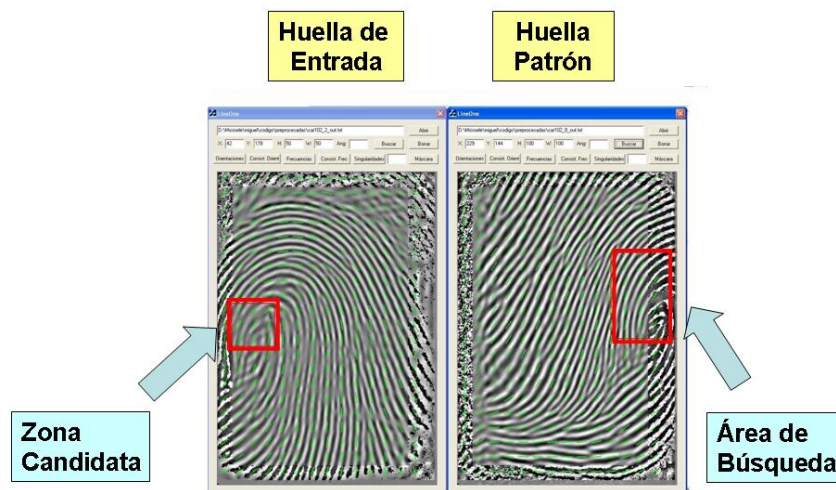


Figura 3.4. Selección de zonas

El número de regiones candidatas y su tamaño, así como el tamaño del área de búsqueda se han determinado experimentalmente para este algoritmo y se detallará en los resultados experimentales, apartado 3.4.

Gracias al desplazamiento y rotación aplicados sobre las regiones candidatas en el cálculo de la correlación sobre el área de búsqueda, es posible refinar la estimación de desplazamiento y rotación realizada en la etapa de alineamiento. Este proceso se ilustra en la Figura 3.5. Tal y como muestra la Figura 3.5, se selecciona una zona candidata (color rosa) en la huella de entrada y un área de búsqueda (color amarillo) de mayor tamaño en la huella patrón. Para evitar los efectos globales producidos por la distorsión no lineal se realizará el cálculo de la correlación de forma local, dividiendo la zona candidata en bloques y calculando la correlación para cada uno de ellos. Para considerar rotaciones y translaciones entre las diferentes muestras es necesario incluir en los cálculos las rotaciones y traslaciones de los bloques de la zona candidata respecto al área de búsqueda.

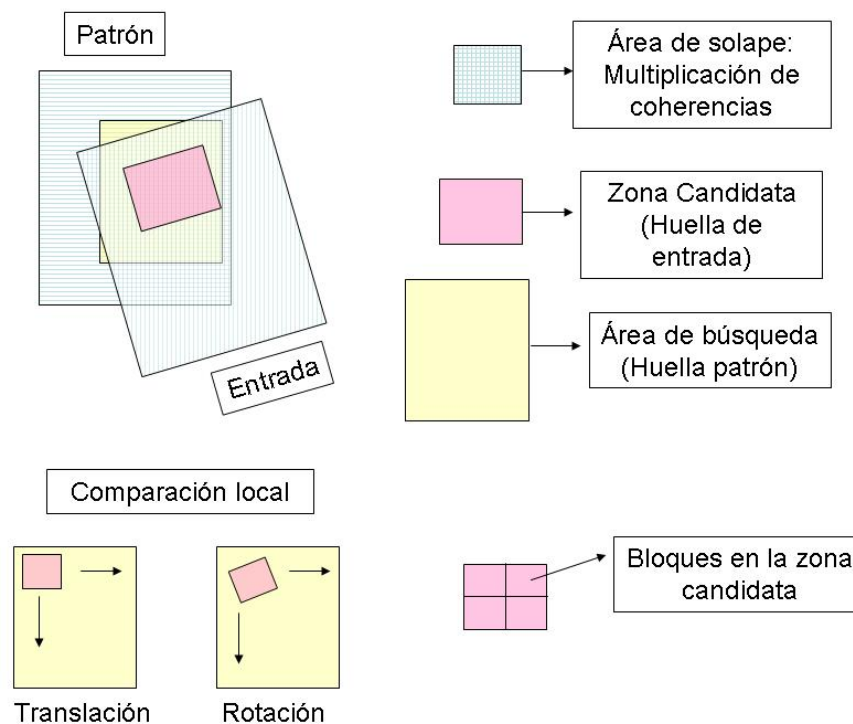


Figura 3.5. Comparación local

3.4- Algoritmo de comparación para imágenes de alta resolución

En la actualidad los sensores de huellas dactilares cada vez proporcionan imágenes con mayor resolución [Parziale_06]. La mayoría de los algoritmos propuestos hasta ahora se han diseñado para imágenes de muy baja resolución y no están adaptados a las posibilidades que ofrecen estos nuevos sensores. Por ejemplo, la conocida técnica de la extracción y comparación de minucias [Maltoni_03], intenta adecuar el reconocimiento automático a la técnica de comparación que realizan los expertos manualmente, identificando los puntos singulares para su posterior comparación. Estas técnicas, cuando se utilizan imágenes de alta resolución, son muy costosas computacionalmente, ya que conllevan un preprocesado exhaustivo para imágenes de grandes dimensiones. Además debido al fuerte preprocesado, las características subyacentes que aparecen en las imágenes de alta resolución desaparecen o se distorsionan debido a los artefactos que genera el preprocesado.

Estudios recientes [Jain_06] muestran que al aumentar la resolución de las huellas aparecen 3 tipos de características: nivel 1 o características macroscópicas, nivel 2 o características microscópicas, y nivel 3. Es para estas últimas para las que no existen métodos de comparación o de extracción adaptados para imágenes de alta resolución. Para características de nivel 1 y 2 existen infinidad de métodos y algoritmos de extracción y comparación, pero la amplia mayoría están adaptados a imágenes de baja resolución. Debido a esta adaptación se pierde gran parte de la información contenida en la huella de alta resolución en el proceso de extracción de características y comparación.

Ya que los sensores de última generación son capaces de mostrar imágenes de huellas con mayor resolución, los algoritmos de extracción y comparación deberán adaptarse a este nuevo escenario y así poder aprovechar toda la información que contiene la huella. La utilización de nuevas técnicas conllevará un aumento de la

precisión y de la eficiencia del sistema. Es por ello que en este apartado se estudiará el análisis de huellas de alta resolución y la extracción de información de las mismas para su posterior comparación mediante técnicas de correlación (apartado 3.3.1). A partir de este estudio se detallará el algoritmo de comparación para huellas de alta resolución y sus diferentes etapas: selección de zonas (apartado 3.3.2) y comparación (apartado 3.3.3).

3.3.1- Selección de la información dentro de la huella

En esta tesis se propone la transformada wavelet [Gonzales_08] como vía para la selección de la información dentro de la huella. La transformada wavelet ha sido utilizada para la compresión de imágenes haciendo posible la detección de información redundante. Esta propiedad se ha aplicado a las huellas dactilares para facilitar la compresión de las imágenes [Walker_99]. Otros trabajos han propuesto la utilización de la transformada wavelet para preprocesado de huellas dactilares [Hatami_05], [Miao_05], [Wei_02], [Xinge_04] y para detección de ser vivo [Moon_05], [Shuckers_04]. Numerosos trabajos se han centrado en la utilización de la transformada wavelet para la extracción de un vector de características de las huellas dactilares [Tico_01_a], [Tico_01_b], [Mokju_04], [Yik_04]. Todos estos trabajos demuestran la capacidad intrínseca de la transformada wavelet para la clasificación de la información contenida en una huella dactilar.

La transformada wavelet es una descomposición jerárquica con subbandas espaciadas logarítmicamente en frecuencia [Walker_99]. La transformada wavelet diádica de una imagen se divide en 4 bandas de coeficientes para cada nivel. Estas bandas son: baja frecuencia (LL), frecuencias medias (LH y HL) y alta frecuencia (HH). Para niveles sucesivos se realiza la misma transformación sobre la banda de baja frecuencia. Los filtros aplicados en cada banda varían en función de la familia wavelet madre utilizada. En la Figura 3.6 se muestra la transformada wavelet diádica de nivel 1 y 2 de una huella dactilar.

A partir de los estudios previos en este campo se ha buscado en esta tesis un método que facilitase la catalogación de la información contenida en las imágenes de huellas dactilares. La finalidad de este estudio es detectar zonas relevantes dentro de la huella para ser utilizadas en el proceso de comparación. Para ello se realizó la transformada wavelet y se estudió el comportamiento estadístico de los coeficientes en las diferentes subbandas de descomposición [Lindoso_07_c]. Para facilitar el análisis de los resultados, las huellas se dividieron en ventanas de tamaño fijo (W x H) y para cada nivel de la transformada wavelet los coeficientes en cada banda se asociaron a la ventana de la imagen original de la que fueron extraídos.

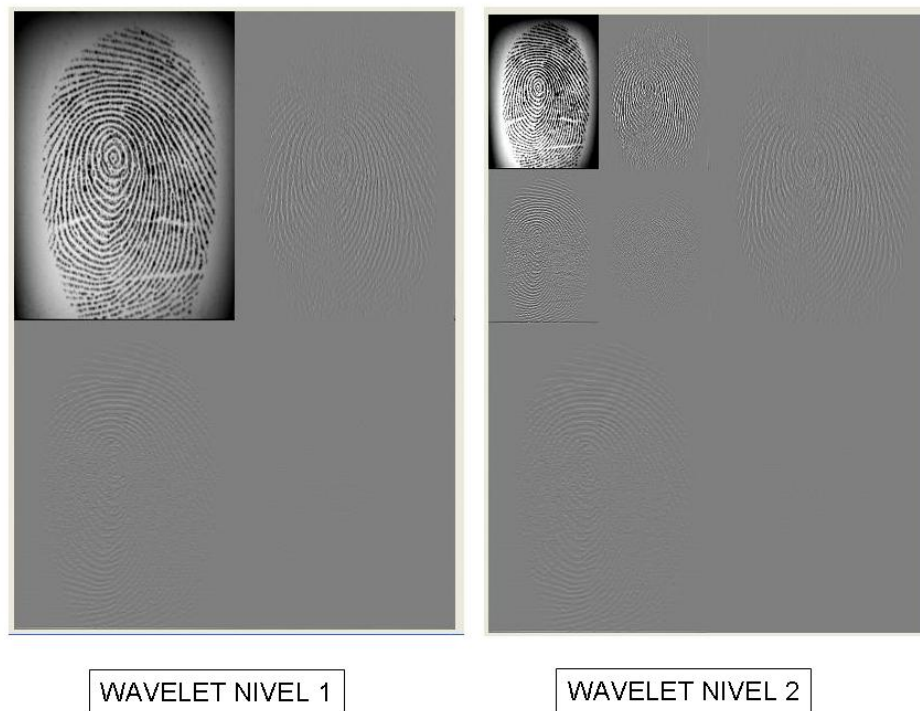


Figura 3.6. Transformada wavelet nivel 1 y 2

Los estadísticos considerados fueron: media, energía, desviación standard y entropía. A continuación se resumen las fórmulas utilizadas:

$$\text{Media: } \frac{\sum C(i, j)}{W \cdot H} \quad (3.16)$$

$$\text{Energía: } \frac{\sum (C(i, j))^2}{W \cdot H} \quad (3.17)$$

$$\text{Desviación Standard: } \frac{\sum (Media - C(i, j))^2}{W \cdot H} \quad (3.18)$$

$$\text{Entropía: } \frac{-\sum prob(C(i, j)) \cdot \log_2(C(i, j))}{W \cdot H} \quad (3.19)$$

En las fórmulas 3.16-3.19, $C(i, j)$ son los coeficientes de la transformada wavelet y W y H son respectivamente la anchura y altura de las ventanas utilizadas para realizar los estadísticos. Estos estadísticos se realizaron para todas las bandas de los tres primeros niveles de la transformada wavelet.

Se realizó un experimento para analizar la información contenida en las imágenes con dos bases de datos: FVC 2006 DB2 B [FVC2006] (10 dedos, 12 muestras por dedo) y una base de datos propia tomada con el sensor FX3000 de Biométrica [FX3000] (7 dedos, 6 muestras por dedo). La resolución de ambas bases de datos es de 569 dpi.

Para analizar los resultados se desarrolló una aplicación visual que muestra la magnitud de los estadísticos junto con la huella original. La huella se divide en ventanas en las que se asigna un valor al estadístico a partir de los valores del mismo en cada banda. Estas ventanas de visualización contienen una suma ponderada del valor del estadístico en cada banda de cada nivel para los píxeles comprendidos en la ventana. El peso de cada banda en la ponderación se asignó de manera independiente. El análisis de los resultados se realizó visualmente mediante la asignación de una paleta de color a los resultados obtenidos.

De los estadísticos considerados, la media y la energía no mostraron buenos resultados para delimitar zonas dentro de la huella. Los mejores resultados para selección de zonas óptimas para la comparación se obtuvieron para transformada

wavelet nivel 2 Daubechies con la desviación standard y considerando únicamente las bandas de frecuencia intermedias de ambos niveles (LH1, HL1, LH2 y HL2). De la exploración visual también se llegó a la conclusión de que la inclusión del nivel 3 reducía la precisión en la selección de zonas. Este mismo efecto se observa utilizando únicamente los coeficientes de nivel 1.

La Figura 3.7 muestra la selección de zonas para diferentes muestras de un mismo dedo de la base de datos propia, y las Figuras 3.8 y 3.9 muestran la selección de zonas realizada en varias muestras de dos dedos diferentes del FVC 2006.

En las Figuras 3.7-3.9 los bloques con color más fuerte (rojo) son los bloques con mayor magnitud, y los bloques más claros (azul) son los de menor magnitud. Tal y como se muestra en las figuras, para todas las huellas se aprecia que el núcleo (“*core*”) se encuentra dentro de la zona de mayor magnitud (zona roja). Aunque el método se muestra efectivo incluso con imágenes de baja calidad, la zona delimitada se muestra demasiado variable y poco homogénea como para utilizarla en un algoritmo de comparación. Ya que el núcleo aparece dentro de esta zona, pero no se puede realizar su localización exacta con el método descrito, en el siguiente apartado se expone un método para la localización del núcleo de manera más precisa mediante transformada wavelet.

3. COMPARACIÓN DE HUELLAS DACTILARES MEDIANTE TÉCNICAS DE CORRELACIÓN

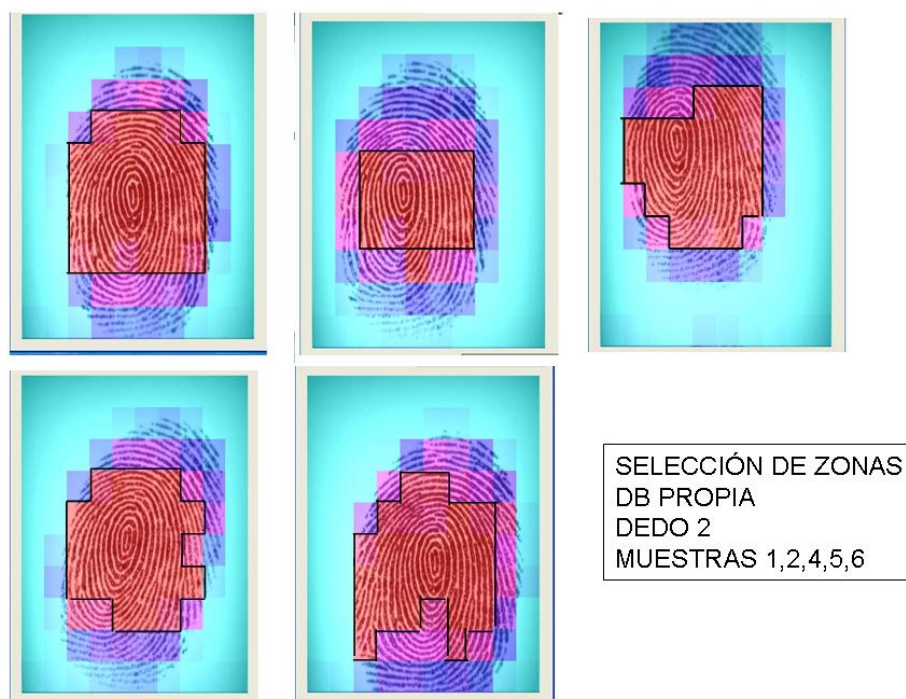


Figura 3.7. Selección de zonas DB propia

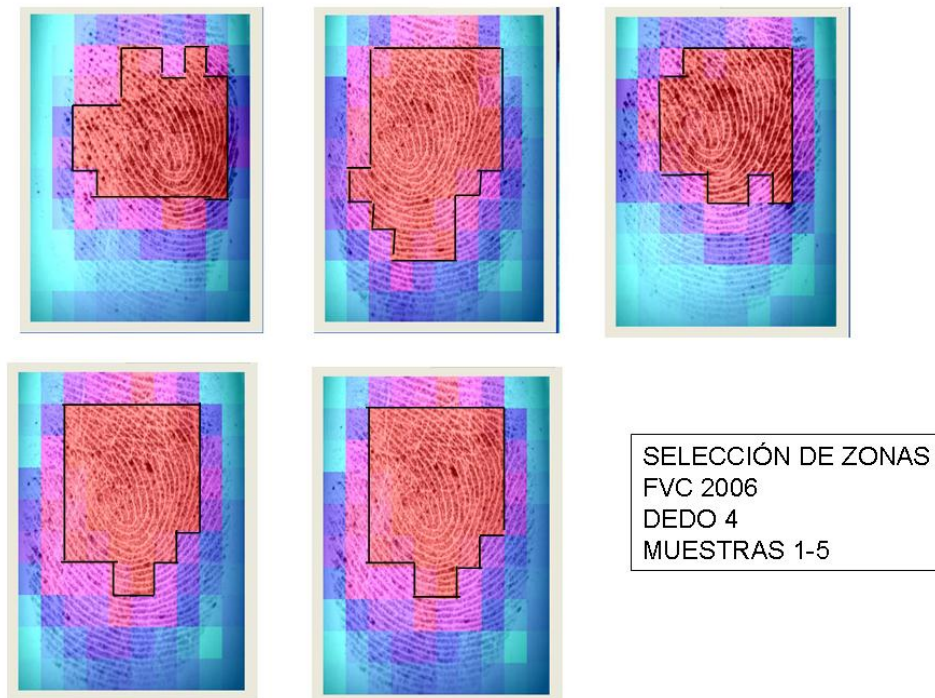


Figura 3.8. Selección de zonas FVC 2006(1)



Figura 3.9. Selección de zonas FVC 2006(2)

A partir del estudio de los estadísticos realizado se apreció también la posibilidad de realizar etapas de preprocesado mediante la transformada wavelet. Para realizar segmentación (delimitación del área de la imagen que contiene huella dactilar) se obtuvieron muy buenos resultados. Se realizaron varios experimentos con diversos estadísticos, niveles de transformada wavelet y bandas. Los mejores resultados se obtuvieron con la entropía y la familia utilizada fue Daubechies. Se observa una mejor discriminación fondo-huella al aumentar el nivel de la transformada wavelet. Sin embargo a partir del nivel 3 esta mejoría se estabiliza y no se aprecian cambios significativos para niveles superiores al 3. Considerando todas las bandas intermedias de los tres primeros niveles de la transformada wavelet (LH1, HL1, LH2, HL2, LH3, HL3) se obtiene una segmentación correcta para todas las huellas de ambas bases de datos. Si se incluyesen el resto de las bandas, la precisión de la segmentación disminuiría.

Los resultados para segmentación obtenidos para una muestra de cada dedo (entropía con todas las bandas intermedias de los tres primeros niveles transformada

wavelet) tanto para la DB propia como para la FVC2006 se muestran en las Figuras 3.10 y 3.11 respectivamente. En estas figuras la paleta de color es inversa a las Figuras 3.7-3.9 y por tanto el color azul claro es el máximo (zona que contiene la huella) mientras que el color rojo es el mínimo. Tal y como muestran las Figuras 3.10 y 3.11, para todas las muestras de ambas bases de datos prácticamente la totalidad de la huella presenta magnitud máxima (azul claro), realizándose por tanto una buena segmentación. En los bordes existe un descenso en la magnitud y por tanto en la escala de color (azul oscuro, rosa). Este descenso se debe a que los bloques considerados tienen un tamaño fijo independientemente de la huella y la zona a tratar. Por lo tanto en áreas fronterizas huella-fondo los bloques abarcan partes de huella y de fondo, haciendo imprecisa la medida en estas zonas. Para mejorar la precisión en los bordes habría que subdividir los bloques en la zona de transición de la magnitud máxima a la mínima.



Figura 3.10. Segmentación DB propia

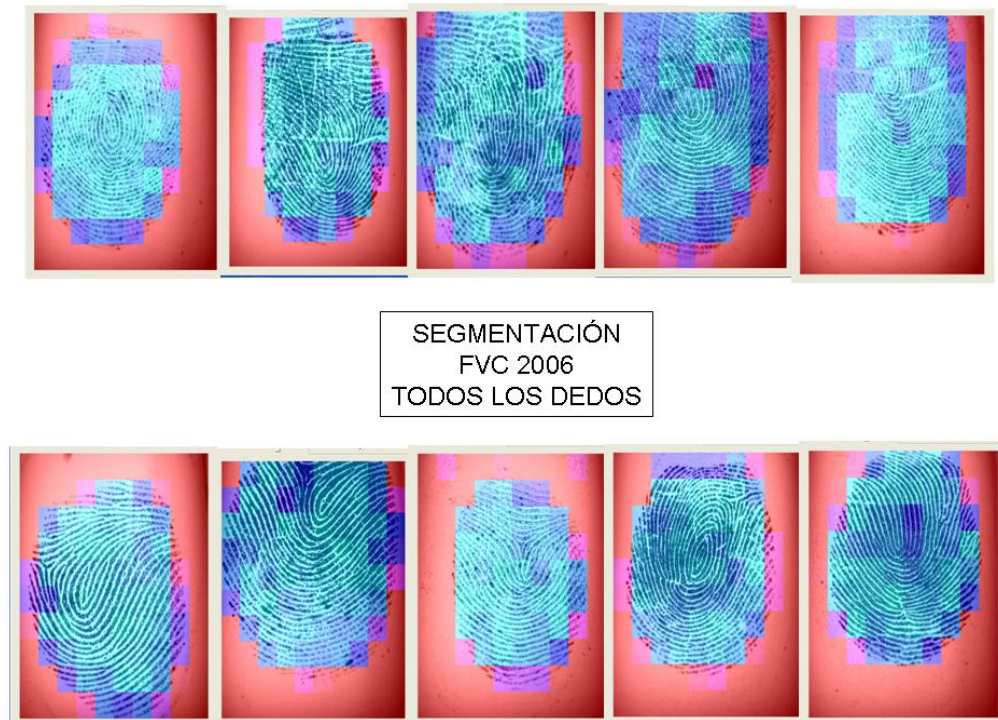


Figura 3.11. Segmentación FVC 2006

3.3.2- Selección de zonas

A partir de los resultados expuestos en el apartado anterior, puede afirmarse que la transformada wavelet se puede utilizar para clasificar la información contenida en la huella. Los resultados mostrados en las Figuras 3.7-3.11 muestran que es posible delimitar zonas distintivas dentro de las huellas y que estas zonas siempre contienen el núcleo o core. El problema que presenta el método descrito anteriormente es la poca homogeneidad que presentan las zonas para su utilización en algoritmos de comparación. Las zonas tienen tamaños variables y no tienen por qué contener regiones coincidentes de una misma huella. En base a esta selección de la información realizada se buscó una manera más homogénea de seleccionar una zona dentro de la huella para su posterior comparación mediante técnicas de correlación [Lindoso_07_d]. Ya que las zonas seleccionadas contenían el núcleo, se

buscó un refinamiento en la localización del mismo, para poder luego seleccionar a partir de un punto concreto una zona de tamaño fijo que lo contenga y dar homogeneidad al método.

El núcleo es una zona muy distintiva de la huella y por tanto ideal para el tipo de comparación que se quiere realizar. El problema que presenta es que suele verse afectado por muchos efectos no deseables, distorsión, ruido, etc. En este caso debido a la mayor calidad del sensor utilizado y a la mayor resolución de las imágenes estos efectos afectan en menor medida al error cometido, como se expondrá en el apartado 3.4.

A partir de los resultados presentados en el apartado anterior se realizó una exploración visual y matemática del comportamiento de las componentes y las familias en el proceso de la transformada wavelet, así como la variación de los estadísticos para tratar de refinar la localización del núcleo. Mediante la observación visual del comportamiento de las componentes de la transformada wavelet se observó una predominancia de ciertas componentes dependiendo de la dirección de las crestas dentro de la huella. Esto permitía delimitar dentro de la huella diferentes zonas con componentes predominantes. Estas zonas presentaban una transición que coincidía con la posición del núcleo y por tanto servían para su localización. Para caracterizar este comportamiento, las huellas se han analizado considerando únicamente la predominancia de las direcciones vertical y horizontal. El resultado para dos huellas diferentes se muestra en la Figura 3.12.

La Figura 3.12 muestra que en la zona que se encuentra en la parte superior del núcleo la orientación predominante (H/V) es horizontal (por debajo de 45°), mientras que en los laterales la orientación predominante es vertical (por encima de 45°). En la parte inferior de la huella, el comportamiento depende del tipo de huella, aunque normalmente aparece una componente horizontal. A través de la inspección visual de esta orientación predominante (H/V) se observó que cualquier tipo de huella presenta componentes horizontales en la zona inmediatamente superior al núcleo. Además estas componentes presentan siempre una forma triangular característica, con un vértice apuntado al núcleo. Este patrón triangular está presente para cualquier tipo de huella, en cualquier posición y bajo cualquier tipo de

efecto adverso como puede ser la distorsión.



Figura 3.12. Orientación predominante (H/V)

A partir de estos descubrimientos se realizó un algoritmo de detección de la localización del núcleo mediante la transformada wavelet que consta de los siguientes pasos: transformada wavelet diádica, determinación de la componente predominante (H/V) y búsqueda del núcleo.

El primer paso es la transformada wavelet diádica de la huella dactilar. El máximo número de niveles que pueden realizarse dependerá de la resolución de la imagen utilizada. No obstante, cabe destacar que las huellas son patrones oscilatorios que acumulan su energía en bandas intermedias y por tanto, si se utilizan demasiados niveles, se perderá información. Es por tanto necesario determinar el número de niveles en función de la resolución de las huellas utilizadas.

En el segundo paso se calcula la componente predominante (H/V). La componente predominante (H/V) imagen es una imagen de orientación que determina la orientación predominante de las crestas considerando únicamente dos

direcciones: horizontal y vertical. En la Figura 3.13 se muestra el proceso que se realiza en este paso del algoritmo.

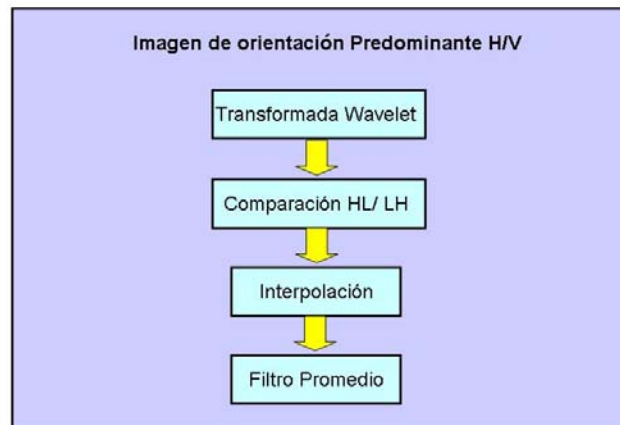


Figura 3.13. Cálculo de la componente predominante H/V

En primer lugar, se analizan las bandas de coeficientes LH y HL en cada nivel de la descomposición de la transformada wavelet. Considerando únicamente esos dos conjuntos de coeficientes se determina qué componente es predominante en cada píxel. Como la transformada wavelet precisa submuestreo, es necesario realizar interpolación para asociar el resultado obtenido a la imagen original. Por último se realiza un filtrado promedio para reducir el ruido introducido por la interpolación y para suavizar el resultado.

El resultado muestra una imagen con la componente predominante en cada píxel de la huella. En la Figura 3.14 se muestra el campo de orientación predominante (H/V) para diversas huellas de la DB2 del FVC2006 [FVC2006]. En la Figura 3.14 se ha asignado a la componente horizontal el valor '1', azul, y a la vertical el valor '0'.

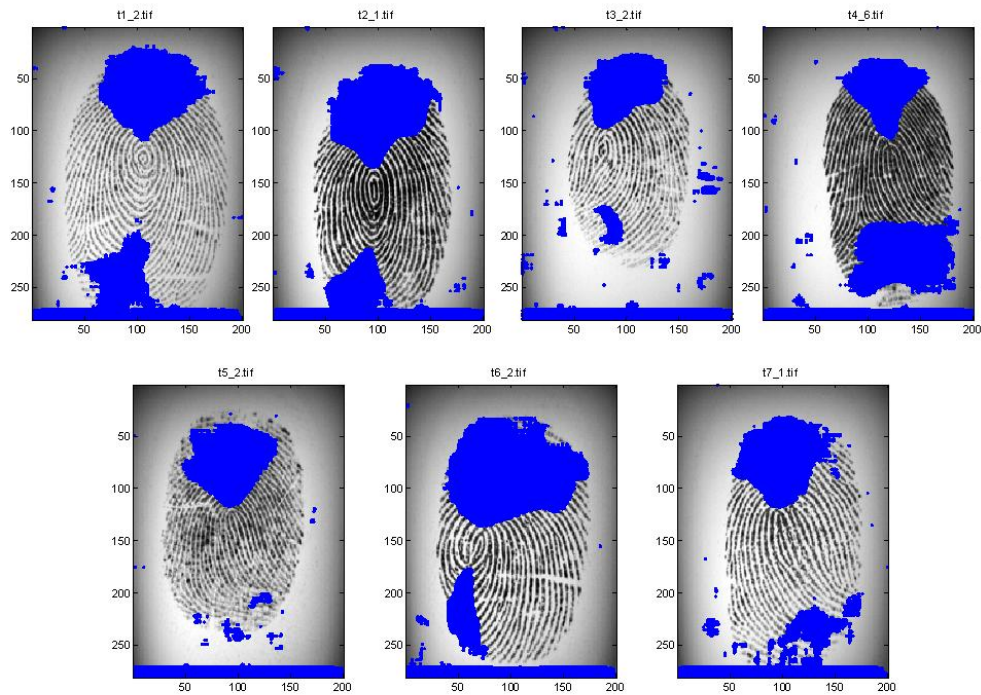


Figura 3.14. Componente predominante H/V (FVC 2006)

Como se puede observar en la Figura 3.14, para todas las huellas existe un patrón triangular en la parte inmediatamente superior al núcleo, con uno de sus vértices apuntando a la posición del núcleo. Este patrón aparece en todos los casos incluyendo huellas de distintos tipos, huellas desplazadas, rotadas, distorsionadas, etc.

En el último paso del algoritmo, a partir de la componente predominante (H/V) se realiza la búsqueda del patrón triangular para poder con ello detectar la localización del núcleo. Para ello es necesario localizar el vértice inferior del patrón triangular generado gracias a la componente predominante (H/V). El proceso se ilustra en la Figura 3.15.

El primer paso para localizar el núcleo es sumar por filas la imagen de componente predominante (H/V) y normalizar el resultado. Gracias a esta operación, se crea un patrón unidimensional vertical que contiene un mínimo local en la zona del núcleo (un ejemplo de este vector unidimensional se muestra en la Figura 3.15). La detección de la posición vertical del núcleo se realiza recorriendo

los resultados del vector desde el borde superior de la imagen. Una vez superado el valor medio del vector unidimensional, se elige como posición vertical del núcleo el primer punto que sobrepase un umbral mínimo preestablecido.

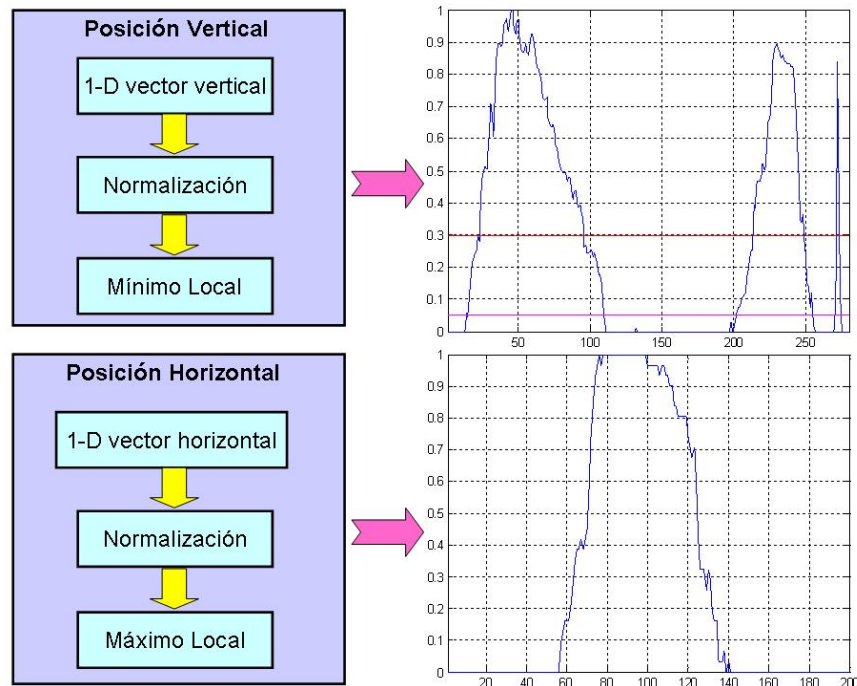


Figura 3.15. Proceso para la detección del núcleo

Una vez obtenida la posición vertical del núcleo, se selecciona la zona de la imagen inmediatamente superior a este punto para determinar la posición horizontal. Para ello se suma por columnas la componente predominante H/V de esa zona inmediatamente superior al núcleo, generando un vector unidimensional horizontal. Dentro de este vector la posición horizontal del núcleo se encontrará en un máximo local. Por lo tanto la posición que contenga un valor máximo se selecciona como posición horizontal del núcleo. Para huellas en las que exista más de una posición con valor máximo se elegirá como posición horizontal del núcleo la media de las posiciones máximas. En ambas direcciones vertical/horizontal se han utilizado umbrales dinámicos. Este proceso se resume en la Figura 3.15.

En la Figura 3.16 se muestran diversas huellas de la DB2 FVC2006 en las que se ha localizado el núcleo mediante el método descrito. Como se aprecia en la

Figura 3.16 el método propuesto detecta correctamente el núcleo incluso en huellas con muy baja calidad.

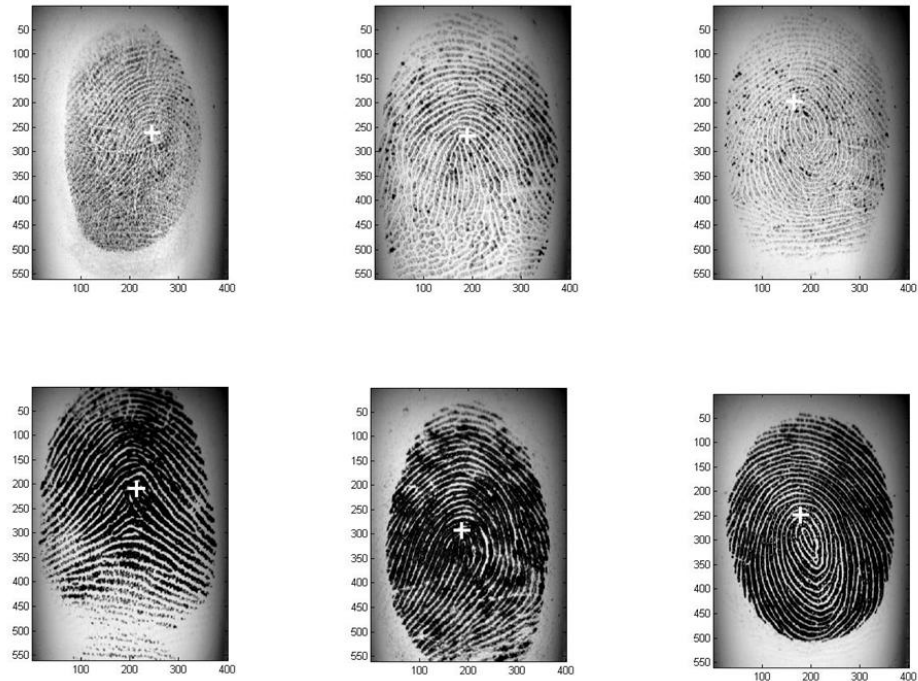


Figura 3.16. Detección del núcleo (FVC 2006)

Los detalles concernientes a las pruebas realizadas para la determinación del método así como la precisión del mismo se detallan en el apartado 3.4.

3.3.3- Comparación

A partir de la posición del núcleo se selecciona para cada huella una zona de tamaño fijo que lo contenga. Una vez seleccionadas las zonas a comparar en ambas huellas se efectúa sobre ellas el cálculo de ZNCC con desplazamiento y rotación (ecuación 3.3).

Cabe destacar que al igual que en el caso del algoritmo de baja resolución detallado en el apartado 3.2, el área de búsqueda (huella patrón) tiene un tamaño

mayor que la zona seleccionada (huella de entrada). En este algoritmo solamente se empleará una zona de la huella de entrada. Sin embargo, para evitar la distorsión no lineal se subdividirá en 4 partes que se correlarán de manera independiente incluyendo traslaciones y rotaciones. Los resultados concernientes al proceso de comparación realizado se detallan en el apartado 3.4.

En la Figura 3.17 se muestra una huella con el área de búsqueda y el núcleo determinados por el algoritmo propuesto.



Figura 3.17. Área de búsqueda y posición del núcleo

3.4- Resultados experimentales

3.4.1- Algoritmo de comparación para huellas de baja resolución

El algoritmo descrito en el apartado 3.2 se ha probado con la base de datos FVC 2000 2 A [Maio_00]. Esta base de datos contiene 8 muestras de 100 dedos diferentes, es decir 800 huellas en total. El tamaño de las imágenes es de 256x364 píxeles.

Para este tamaño de huella se ha calculado el campo de orientación con un tamaño de 18x24.

El número de zonas seleccionadas en la huella de entrada son 3 de tamaño 50x50 píxeles. El tamaño del área de búsqueda (en la imagen patrón) es de 100x100 píxeles. Tanto los tamaños de ambas zonas como el número de zonas seleccionadas se determinaron empíricamente gracias a un experimento inicial con una muestra reducida de huellas de la base de datos. En la etapa de comparación se han realizado 13 rotaciones en incrementos de dos grados (6 en cada sentido de las agujas del reloj) sobre la rotación calculada en la etapa de alineamiento.

Los experimentos se han llevado a cabo siguiendo las premisas del FVC [Wayman_05]. Para determinar el FMR ("False Match Rate") la primera muestra de cada dedo se ha comparado con todas las primeras muestras del resto de los dedos. Para determinar el FNMR ("False Non Match Rate") se han comparado todas las muestras de cada dedo. El ROC ("Receiver Operating Characteristic curve") que se ha obtenido para el algoritmo se muestra en la siguiente figura. El EER ("Equal Error Rate") obtenido es 8%.

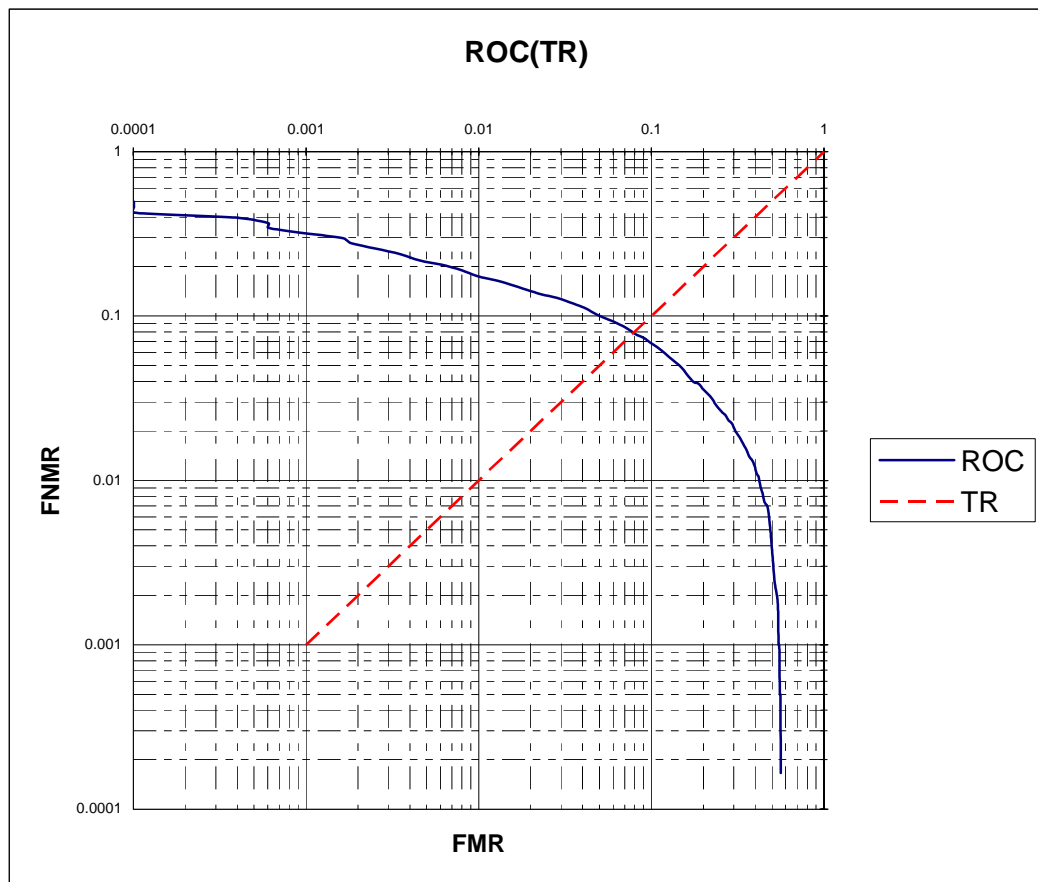


Figura 3.18. ROC en escala logarítmica para el algoritmo de comparación para imágenes de baja resolución para la base de datos FVC 2000 2 A

La mayoría de los errores son debidos a la mala calidad de las huellas o a un solapamiento insuficiente entre las huellas de entrada y patrón. Estos casos podrían solventarse con una etapa de preprocesado adicional. Las regiones que se han utilizado contienen una parte significativa de la huella. Esto ocasiona que en ciertos casos no se pueda evitar que las regiones contengan zonas de baja calidad de imagen o incluso que las regiones utilizadas no solapen completamente. Estos casos podrían detectarse en la etapa de preprocesado o incluso después mediante la utilización de umbrales para las etapas de alineamiento y comparación. Ejemplos de este tipo de imágenes, pertenecientes a la base de datos utilizada, se muestra en la Figura 3.19.



Figura 3.19. Huellas de baja calidad

En una aplicación práctica deberían medirse la calidad y el solapamiento de las muestras. Con estas medidas se puede comprobar si la muestra capturada cumple unos requisitos mínimos de solapamiento y calidad de imagen. En el caso de no cumplir estos requisitos mínimos la muestra debería ser rechazada, solicitándose al usuario una nueva muestra. Sin embargo en las pruebas realizadas no se ha considerado la posibilidad de rechazar las muestras de baja calidad o con bajo solapamiento.

El tiempo necesario para realizar una verificación de una huella es de 1.15 segundos: 0.25 segundos para el preprocesado y 0.9 segundos para la comparación. Estas medidas se han realizado para una implementación en C del algoritmo con un PC Pentium IV a 3 GHz con 2 GB de RAM.

La implementación del algoritmo podría optimizarse. Hay que destacar que dados un conjunto de parámetros (tamaño del campo de orientación, tamaño de la zona de búsqueda, tamaño de las regiones, etc.) el tiempo de cálculo es

independiente de las características de la huella considerada, ya que siempre se realizan todos los cálculos en todas las comparaciones. Por lo tanto el esfuerzo computacional podría reducirse imponiendo umbrales para todas las etapas y emitiendo una decisión tan pronto como sea posible.

3.4.2- Algoritmo de comparación para huellas de alta resolución

3.4.2.1- Detección del núcleo

El método de detección del núcleo ha sido probado con la base de datos FVC 2006 DB2 [FVC2006]. Esta base de datos contiene imágenes de 560x400 píxeles de 140 dedos con 12 muestras por dedo, siendo en total 1680 huellas.

La transformada wavelet diádica se realizó con diversas familias, llegando a la conclusión de que no afectaba en demasía el cambio de unas a otras para este método. Por lo tanto se seleccionó la familia Daubechies 4, ya que en numerosos trabajos de aplicación de la transformada wavelet a las huellas dactilares se utiliza esta familia. El número de coeficientes de la familia tampoco afecta al método, por lo que se escogió uno intermedio.

Para calibrar la detección del núcleo se seleccionó una muestra de esta base de datos consistente en 12 muestras de 10 dedos (120 huellas). Para la calibración se utilizaron diferentes niveles de descomposición de la transformada wavelet. A partir de estos experimentos se determinó que se puede utilizar hasta nivel 3 con las imágenes de esta base de datos. Sin embargo, los niveles 1 y 2 mostraron una mejor definición en la formación del patrón triangular. En los experimentos se observó que es necesario aumentar el tamaño del filtro promedio a medida que el nivel de descomposición de la transformada wavelet aumenta y además su tamaño influye considerablemente en el resultado final. Los mejores resultados se obtuvieron para

nivel uno con filtrado promedio de 15x15 y para nivel 2 con filtrado promedio de 17x17.

Para obtener una primera aproximación de la precisión del algoritmo de detección propuesto se realizó una detección manual del núcleo de cada huella de la base de datos y se comparó visualmente con la posición obtenida con el algoritmo de detección. Utilizando la base de datos completa, el error de detección del núcleo es 6.67% (112 huellas) para descomposición wavelet de nivel 1 y ventana de promediado de 15x15. Si se considera nivel 2 y una ventana de promediado de 17x17 el error es de 3.57% (60 huellas). En ambos experimentos se ha aplicado un preprocesado ligero a las huellas previamente a la detección. Este preprocesado consiste únicamente en segmentación y ecualización del histograma. Sin embargo, cabe destacar que incluso sin ecualización de histograma el método de detección es efectivo mostrando un error de 4.35% (73 huellas) para descomposición wavelet de nivel 2 y ventana de promediado de 17x17. El efecto del segmentado es prácticamente inapreciable, porque sólo afecta a un pequeño subconjunto de huellas. En la Tabla 3.1 se muestra un resumen de los resultados obtenidos.

	Errores
Nivel 1 Ventana 15x15 Con preprocesado	6.67% (112 huellas)
Nivel 2 Ventana 17x17 Sin preprocesado	4.35 % (73 huellas)
Nivel 2 Ventana 17x17 Con preprocesado	3.57 % (60 huellas)

Tabla 3.1. Errores en la localización del núcleo para la FVC2006 DB2 (1680 huellas)

Para calibrar con mayor exactitud la precisión del método propuesto se ha realizado otro experimento con la muestra de calibración (120 huellas). En este experimento se midieron y analizaron numéricamente los errores entre la posición del núcleo marcada manualmente y la detectada. En la posición horizontal el error era menor de 25 píxeles para el 92.5% de las huellas mientras que en la posición vertical el error era menor de 25 píxeles para el 97.5% de las huellas. Considerando estos dos errores como errores máximos en ambas direcciones, la máxima distancia euclídea entre el núcleo marcado manualmente y el detectado es de 39 píxeles. Teniendo en cuenta que para la base de datos utilizada la distancia media entre crestas es de 12 píxeles, el error máximo de localización del núcleo es de 3.5 crestas.

El análisis de los resultados muestra 3 posibles causas de error en la detección:

- 1- Huellas con muy mala calidad.
- 2- Núcleo en el extremo de la imagen.
- 3- Huellas tipo arco con muy baja curvatura.

Ejemplos de estos errores se muestran en la Figura 3.20:

De los 60 errores que muestra el método (para el experimento con preprocesado ligero, descomposición wavelet de nivel 2 y promediado de tamaño 17x17), 32 se deben a localización del núcleo en el extremo superior de la imagen, 20 son ocasionados por huellas de tipo arco con baja curvatura y 8 son debidos a mala calidad de la imagen.

Para corregir estos errores, en el caso de mala calidad de imagen debería llevarse a cabo una etapa de preprocesado adaptada a los problemas que presenten las imágenes. Sin embargo, tal y como muestran los resultados, esto sería efectivo para un número muy reducido de casos (8 huellas que equivalen al 0.5%). Las huellas que presentan su núcleo en el extremo superior de la imagen no muestran el patrón triangular en la imagen de componente predominante H/V. Estas huellas deberían ser detectadas y rechazadas en el proceso de captura. Finalmente, el error debido a huellas de tipo arco con baja curvatura es el más difícil de solucionar para el método propuesto ya que la baja curvatura repercute en la precisión de la

localización del núcleo. Sin embargo, para este tipo de huellas se han detectado patrones específicos que podrían ser detectados refinando el método.

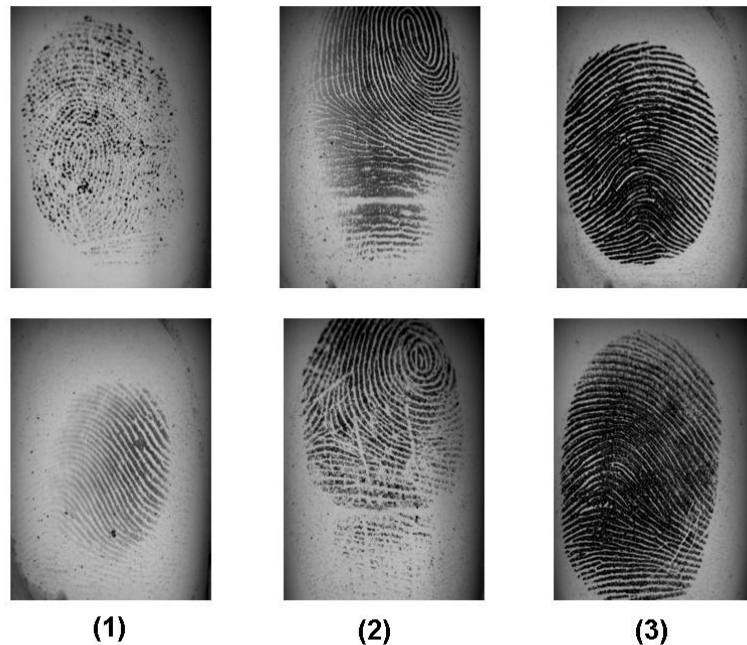


Figura 3.20. Errores de detección del núcleo: (1) Huellas con baja calidad, (2) núcleo en el extremo superior de la imagen, (3) Huellas tipo arco con baja curvatura

El método propuesto es también eficiente. Su implementación en Matlab en un PC Pentium IV (3.2 GHz y 2GB RAM) requiere únicamente 0.118 segundos. Si se incluye preprocesado (segmentación y ecualización del histograma), el tiempo medio es de 0.291 segundos. Si se utiliza una implementación en C en el mismo equipo el tiempo medio es de 54.7 ms.

El método propuesto para la detección del núcleo contrasta con todos los métodos propuestos hoy en día, ya que utiliza directamente la imagen sin preprocesar y sin extraer características como el campo de orientación. La precisión requerida para el proceso de comparación efectuado es más que suficiente ya que se busca una zona que contenga el núcleo. El método propuesto podría refinarse para obtener una localización más cercana a la marcada manualmente.

3.4.2.2- Algoritmo de comparación

Para la comparación se ha utilizado la base de datos de muestra DB2 del FVC2006 utilizada para calibración (10 dedos, 12 muestras) con 120 huellas.

El núcleo se detectó con los parámetros que mostraron mejores resultados según se ha detallado anteriormente: nivel 2 de descomposición de la transformada wavelet, ventana de promediado de 17x17 y preprocesado ligero consistente en segmentación y ecualización de histograma.

A partir de la posición del núcleo se selecciona en ambas huellas una zona que lo contenga. El área de búsqueda seleccionada del patrón es de 150x150 píxeles, mientras que la zona elegida en la imagen de entrada es de 50x50 píxeles.

Los experimentos se llevaron a cabo inicialmente con una zona de la imagen de entrada de 50x50 píxeles sin dividirla en subzonas y se pudo apreciar que los resultados se veían afectados por la distorsión, provocando una bajada sistemática de los resultados en la correlación y por tanto un umbral para el EER muy bajo (en torno al 30%). Para solucionar este problema se decidió dividir la zona seleccionada de la imagen de entrada en 4 zonas de 25x25 píxeles. Cada una de estas subzonas se correla de forma independiente con el área de búsqueda incluyendo rotación y traslación. La rotación considerada comprende el rango de -15° a 15° con incrementos de 1° .

De los resultados independientes obtenidos con las 4 subzonas se puede apreciar una variación considerable en los resultados y por tanto se utilizaron diversas técnicas estadísticas para determinar la mejor manera de fusionar estos resultados independientes. La media de los resultados de las cuatro zonas no es una solución fiable, ya que un resultado alto dentro del conjunto perjudica a las correlaciones negativas y favorece a las positivas. Estudiando las posibles soluciones se llegó a la conclusión de que la solución que mejores resultados muestra es realizar la media entre el valor de correlación máximo y la mediana de las otras tres zonas restantes.

Con esta aproximación el EER es de 9.865% y el umbral de 0.6756, tal y como se muestra en la Figura 3.21. Las huellas consideradas, pertenecientes a la base de datos de calibración (120 huellas), presentan 4 errores en la detección del core, provocando 4 errores en el alineamiento de las huellas, es decir el 3.3% de los casos.

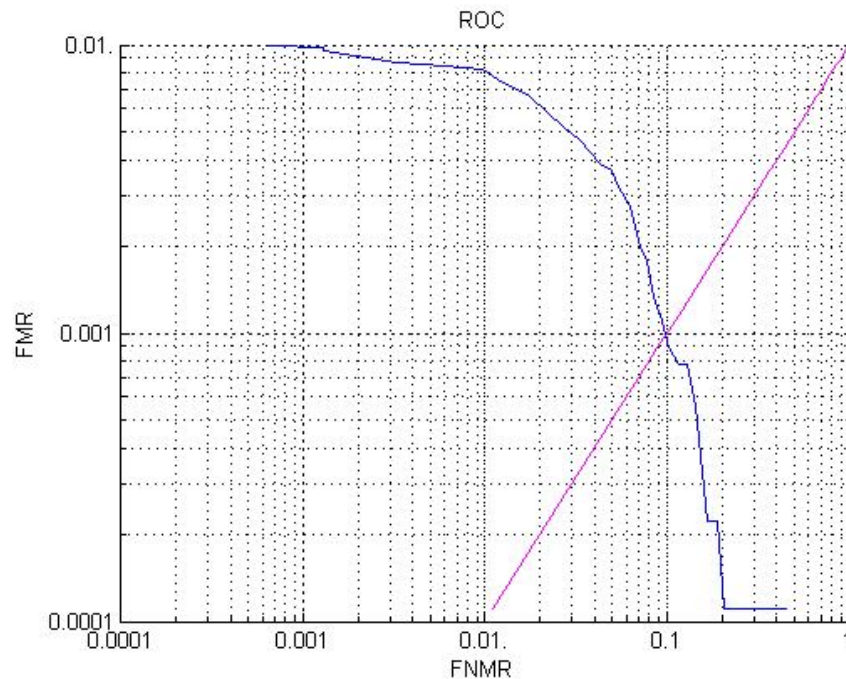


Figura 3.21. ROC en escala logarítmica para el algoritmo de alta resolución para la base de datos de calibración del FVC2006. (120 huellas)

Con esta misma aproximación se realizó el experimento con toda la DB2 del FVC2006, 1680 huellas. El resultado se muestra en la Figura 3.22, el EER obtenido es de 15 %. Este aumento del error se debe a la existencia de huellas con baja calidad y elevada distorsión. Estos efectos provocan errores en la etapa de comparación. El método propuesto podría mejorar en precisión si se solventasen estos dos problemas mediante un preprocesado específico.

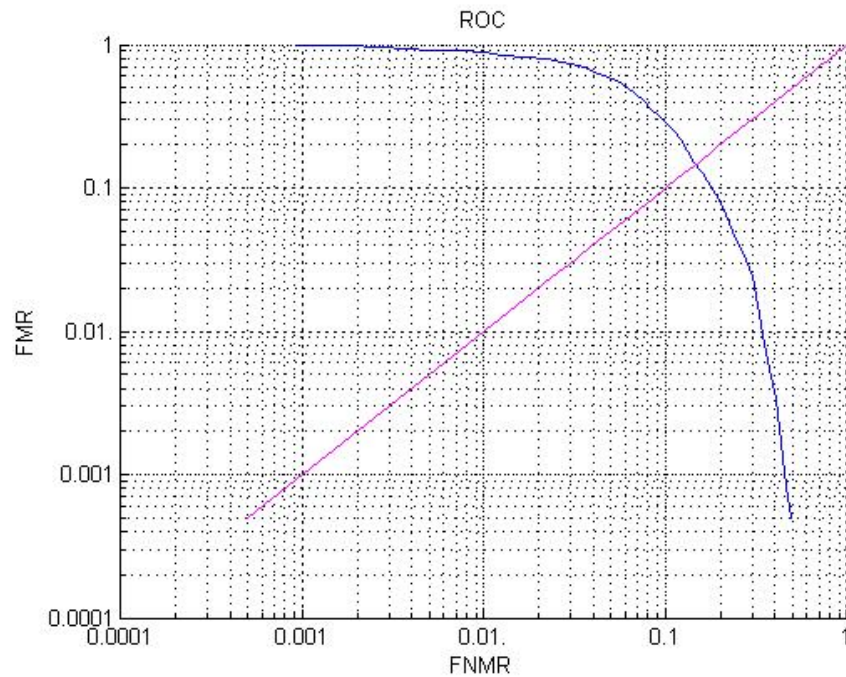


Figura 3.22. ROC en escala logarítmica para el algoritmo de alta resolución para DB2 FVC2006. (1680 huellas)

Una implementación en C del algoritmo propuesto en un PC Pentium IV (3.2 GHz y 2GB RAM) tiene un rendimiento de 54.7 ms para la selección de zona y 1.350 s para la comparación, resultando en un tiempo total de 1.4047 segundos.

3.5- Conclusiones

En este capítulo se han presentado dos algoritmos de comparación para imágenes de baja y alta resolución basados en técnicas de correlación que son aportación original de esta tesis.

El algoritmo para imágenes de baja resolución consta de las siguientes etapas: preprocesado, alineamiento, selección y comparación. En el alineamiento se correlan las orientaciones del campo de orientación de ambas huellas para estimar la orientación y el desplazamiento. Una vez determinados estos parámetros se realiza una selección de las zonas distintivas y con buena calidad de imagen en las que se realizará el cálculo de la correlación. Para ello se utiliza la coherencia de la orientación de ambas imágenes teniendo en cuenta el desplazamiento y orientación ya calculados. Una vez seleccionadas las zonas, se calcula ZNCC volviendo a realizar desplazamientos y rotaciones para refinar la estimación obtenida en la etapa de alineamiento.

Los resultados experimentales demuestran que el algoritmo para imágenes de baja resolución es preciso con un EER del 8% sobre una base de datos de concurso. Además, este algoritmo utiliza una nueva técnica de alineamiento basada en la correlación del campo de orientación. Este método de alineamiento es también una aportación original de esta tesis. Por otra parte, este algoritmo utiliza técnicas de correlación en todas sus etapas y es por tanto adecuado para conseguir una buena aceleración hardware.

El algoritmo para imágenes de alta resolución consiste en dos etapas: detección del núcleo y comparación. La detección del núcleo se realiza directamente sobre la imagen sin preprocesar mediante la transformada wavelet. A partir de esa posición se selecciona tanto el área de búsqueda en el patrón como la zona candidata en la huella de entrada. Para evitar la distorsión no lineal, la zona candidata se divide en 4 subzonas que se correlan de forma independiente sobre el área de búsqueda

teniendo en cuenta desplazamiento y rotación.

Los resultados experimentales demuestran que el algoritmo para huellas de alta resolución también es preciso con un EER del 9.865% probado con una base de datos de concurso. Este resultado ha sido obtenido con imágenes sin preprocesar, lo que significa que el método no deforma la información contenida en la imagen introduciendo artefactos. Este algoritmo, al igual que el de imágenes de baja resolución, presenta una técnica novedosa para la detección del núcleo basada en la transformada wavelet, que es aportación original de esta tesis. El algoritmo para imágenes de alta resolución no utiliza la correlación cruzada en todas las etapas, pero cabe destacar que el filtrado bidimensional realizado en la transformada wavelet presenta una marcada similitud con la correlación cruzada. Esto facilita en gran medida su aceleración con arquitecturas similares a las que se utilizan para calcular la correlación cruzada

Ambos algoritmos utilizan técnicas de correlación de imágenes como medida de similitud y por tanto no comparan posición y tipo de características, sino toda la información contenida dentro de la imagen. Los dos algoritmos propuestos son susceptibles de refinamientos en diversas etapas para conseguir mejorar su precisión.

Ambos algoritmos conllevan una elevada carga computacional, común a muchos algoritmos en el ámbito del procesado de imágenes. Sin embargo, debido a la regularidad de las operaciones empleadas, los algoritmos son susceptibles de acelerarse notablemente mediante hardware específico y es por ello que a lo largo de los siguientes capítulos de esta tesis se explorarán soluciones hardware para conseguir alta eficiencia con este tipo de algoritmos.

Capítulo 4:

ACCELERACIÓN HARDWARE DEL CÁLCULO DE LA CORRELACIÓN

4.1-	INTRODUCCIÓN.....	92
4.2-	CORRELACIÓN CRUZADA	94
4.3-	ARQUITECTURA ESPACIAL	98
4.4-	ARQUITECTURA ESPECTRAL	104
4.5-	NORMALIZACIÓN	107
4.6-	RESULTADOS EXPERIMENTALES	110
4.6.1-	<i>Implementación de las arquitecturas y selección de FPGA.....</i>	<i>111</i>
4.6.2-	<i>Consumo de recursos y rendimiento</i>	<i>113</i>
4.6.3-	<i>Aplicación a la comparación de huellas dactilares.</i>	<i>116</i>
4.7-	CONCLUSIONES	125

4.1- Introducción

En el capítulo 3 se presentaron dos algoritmos de comparación de huellas dactilares basados en técnicas de correlación. Ambos algoritmos utilizan la correlación cruzada como medida de la similitud de las imágenes. Dicha operación conlleva una alta carga computacional que repercute negativamente en el rendimiento de los algoritmos. Esta alta carga computacional contrasta con los estrictos requisitos temporales que se requieren en los escenarios actuales, sobre todo aquellos con respuesta en tiempo real.

Para comparar dos imágenes mediante técnicas de correlación es necesario tener en cuenta las características de las imágenes a tratar. En el caso de las huellas dactilares es necesario realizar la comparación por bloques para disminuir en la medida de lo posible efectos que aparecen sobre las huellas de forma global, como pueden ser: distorsión no lineal, cambios de brillo, presión, etc. Este cálculo local conlleva un aumento de la carga computacional ya que es necesario realizar el cálculo de correlación sobre patrones pequeños desplazados sobre una imagen de entrada extensa. El tamaño de los bloques considerados es un parámetro de suma importancia y depende de la resolución de las imágenes consideradas. Las huellas tienen que dividirse en bloques que sean lo suficientemente grandes como para contener información distintiva pero lo suficientemente pequeños como para que la distorsión no lineal no les afecte

Normalmente el cálculo a nivel local no se realiza sobre toda la huella, ya que es muy costoso computacionalmente, y además porque la correlación de zonas no distintivas puede decrementar la precisión de la comparación. Una estrategia que proporciona mejores resultados consiste en calcular la correlación sólo para ciertas regiones previamente seleccionadas de cada huella. Se pueden utilizar regiones alrededor de las minucias [Kovacks_00], [Huvanandana_00] o regiones en las que se localiza el máximo de autocorrelación [Bazen_00]. Otro criterio de selección que proporciona buenos resultados es seleccionar zonas con buena calidad de imagen,

ya que el ruido repercute negativamente en los resultados de la correlación.

Además de realizar el cálculo de forma local es necesario realizar rotaciones y traslaciones de la imagen de entrada respecto al patrón, lo que eleva aún más la carga computacional.

A consecuencia de todo lo expuesto anteriormente, el reconocimiento de huellas dactilares basado en técnicas de correlación requiere un gran número de correlaciones para emitir una decisión. Esto provoca una elevada carga computacional que es de mayor importancia para entornos que requieran respuesta en tiempo real. Aunque existen trabajos que tratan de reducir la carga computacional mediante selección de bloques y rotaciones en los que calcular la correlación [Maltoni_03], [USP_02], el rendimiento que se consigue sigue siendo insuficiente para muchos escenarios.

En este capítulo se estudia la aceleración del cálculo de la correlación cruzada mediante arquitecturas hardware para adecuar su rendimiento a requisitos en tiempo real. Los resultados obtenidos son extrapolables para cualquier otra aplicación que requiera medida de similitud de imágenes o de partes de las mismas. Entre los campos de aplicación en los que se utiliza el cálculo de la correlación cruzada cabe destacar el procesado de imágenes médicas, modelado tridimensional, monitorización de imágenes vía satélite, reconocimiento de patrones, biometría, etc.

Las arquitecturas expuestas en este capítulo pueden utilizarse para acelerar la fase de comparación en sistemas de reconocimiento de huellas dactilares. La reducción en el tiempo de cálculo que se consigue con las arquitecturas hardware permite incluso aumentar el número de zonas, rotaciones y traslaciones en las que se calcula la correlación. Considerando este conjunto más exhaustivo de cálculos de correlación, se puede obtener una precisión mayor en la comparación de huellas dactilares sin que esto implique un aumento del tiempo de cálculo.

Este capítulo consta de un resumen teórico en el apartado 4.2, a partir del cual se fundamentarán teóricamente las arquitecturas que se detallan en los siguientes apartados. Las arquitecturas se subdividen en el dominio del espacio, apartado 4.3, y de la frecuencia, apartado 4.4, adaptando al hardware las dos posibles formas,

equivalentes matemáticamente, de calcular la correlación cruzada. Para la aplicación de la correlación cruzada en el campo de la comparación de imágenes y más concretamente en la comparación de las huellas dactilares es necesaria la normalización del resultado. Arquitecturas específicas para la normalización han sido diseñadas y se detallan en el apartado 4.5. Los resultados experimentales se presentan en el apartado 4.6.

4.2- Correlación cruzada

La correlación cruzada se utiliza ampliamente como medida de similitud en el ámbito del procesamiento de imágenes. Este cálculo se utiliza en la etapa de comparación de los dos algoritmos de comparación de huellas dactilares propuestos en el capítulo 3. Para la comparación de huellas dactilares se utiliza la correlación cruzada normalizada de media cero, ZNCC. Esto es debido a que ZNCC es invariante respecto a transformaciones afines y por tanto permite minimizar las variaciones entre diferentes muestras de una misma huella debidas a cambios de iluminación, presión, elasticidad de la piel, etcétera.

En el apartado 3.2 del capítulo 3 se formuló matemáticamente la correlación cruzada normalizada de media cero [Crouzil_96], ZNCC (ecuación 3.3). ZNCC se define mediante la fórmula 4.1, en la que T es el patrón e I la imagen de entrada desplazada p y q píxeles en las direcciones horizontal y vertical respectivamente.

$$ZNCC(p, q) = \frac{CC(T - \bar{T}, I(p, q) - \bar{I}(p, q))}{\|T - \bar{T}\| \cdot \|I(p, q) - \bar{I}(p, q)\|} = \frac{\sum_{i=0}^{n-1} \sum_{j=0}^{m-1} (T(i, j) - \bar{T})(I(p+i, q+j) - \bar{I}(p, q))}{\sqrt{\sum_{i=0}^{n-1} \sum_{j=0}^{m-1} (T(i, j) - \bar{T})^2} \sqrt{\sum_{i=0}^{n-1} \sum_{j=0}^{m-1} (I(p+i, q+j) - \bar{I}(p, q))^2}} \quad (4.1)$$

La ecuación 4.1 también puede expresarse de la siguiente manera:

$$ZNCC(p, q) = \frac{\sum_{i=0}^{n-1} \sum_{j=0}^{m-1} T(i, j) I(p+i, q+j) - nm \bar{T} \bar{I}(p, q)}{\sqrt{\sum_{i=0}^{n-1} \sum_{j=0}^{m-1} T^2(i, j) - nm \bar{T}^2} \sqrt{\sum_{i=0}^{n-1} \sum_{j=0}^{m-1} I^2(p+i, q+j) - nm \bar{I}^2(p, q)}} \quad (4.2)$$

En la ecuaciones 4.1 y 4.2, $\|T\| = \sqrt{\sum_{i,j} T^2(i, j)}$ es la norma de la imagen patrón T , \bar{T} es la intensidad media de T , $\bar{I}(p, q)$ es la intensidad media de la imagen de entrada I , desplazada p y q píxeles, y nm es el tamaño del patrón (n es la dimensión vertical y m la horizontal).

Normalmente la imagen patrón suele ser procesada antes de la comparación y por tanto en este proceso se llevará a cabo una normalización para que $\bar{T} = 0$ y $\|T\| = \sqrt{nm}$. Teniendo esto en cuenta se puede llegar a la siguiente simplificación para el cálculo de ZNCC [Siegel_82]:

$$ZNCC(p, q) = \frac{\sum_{i=0}^{n-1} \sum_{j=0}^{m-1} T(i, j) I(p+i, q+j)}{\sqrt{nm \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} I^2(p+i, q+j) - n^2 m^2 \bar{I}^2(p, q)}} = \frac{CC(p, q)}{\sqrt{nm SS(p, q) - S(p, q)^2}} \quad (4.3)$$

En la ecuación 4.3 CC es la correlación cruzada, S es la suma de píxeles de la imagen de entrada y SS la suma de cuadrados de los píxeles de la imagen de entrada, en ambos casos considerando desplazamientos. CC , S y SS se definen mediante las siguientes ecuaciones:

$$CC(p, q) = \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} T(i, j) I(p+i, q+j) \quad (4.4)$$

$$S(p, q) = \sum_{i=0}^{i=n-1} \sum_{j=0}^{j=m-1} I(p+i, q+j) \quad (4.5)$$

$$SS(p, q) = \sum_{i=0}^{i=n-1} \sum_{j=0}^{j=m-1} I(p+i, q+j)^2 \quad (4.6)$$

En la búsqueda del máximo de ZNCC se puede evitar el cómputo de la raíz cuadrada y de la división de la ecuación 4.3 mediante la comparación de cuadrados en numerador y denominador, dando lugar a la comparación mostrada en la ecuación 4.8.

En concreto la comparación:

$$ZNCC(p_1, q_1) > ZNCC(p_2, q_2) \quad (4.7)$$

es equivalente a la siguiente expresión:

$$CC^2(p_1, q_1)D^2(p_2, q_2) - CC^2(p_2, q_2)D^2(p_1, q_1) > 0 \quad (4.8)$$

en la cual D(p, q) es el denominador de la ecuación 4.3.

Para incluir la rotación en la ecuación 4.3 se realizará la rotación del patrón y se repetirá el cálculo de ZNCC y la búsqueda del máximo para una serie de patrones rotados. Esto conlleva una elevada carga computacional ya que es necesaria la repetición del cálculo para todas las posibles combinaciones de desplazamiento y rotación de las imágenes. Una posible reducción del espacio de búsqueda consiste en considerar rotación y desplazamiento por separado, buscando primero el máximo de desplazamiento y una vez determinado, calcular CC incluyendo rotaciones únicamente alrededor de ese punto [USP_02], [DeCastro_87].

Otra posible solución para aumentar la eficiencia computacional es realizar el cálculo de CC en el dominio de la frecuencia. Esto es posible gracias al teorema de la correlación [Gonzales_92]. La ecuación 4.4 se transformaría en la siguiente expresión:

$$CC = F^{-1}(F^*(T) * F(I)) \quad (4.9)$$

siendo $F(T)$ y $F(I)$ las transformadas de Fourier del patrón y la imagen de entrada respectivamente, $F^*(T)$ el complejo conjugado de la transformada de Fourier del patrón, T , y F^{-1} la transformada de Fourier inversa. La ecuación 4.9 requiere el cálculo de tres transformadas de Fourier, dos directas y una inversa y una multiplicación compleja conjugada. El resultado será una matriz que contendrá en cada posición $CC(p,q)$ los resultados de CC para el patrón y la imagen de entrada desplazada p y q píxeles en las direcciones vertical y horizontal respectivamente. En este caso se reduce el coste computacional respecto al cálculo de manera directa (ecuación 4.4) debido a que los desplazamientos están incluidos y es posible utilizar FFT (“Fast Fourier Transform”) para el cálculo de las transformadas de Fourier.

Este capítulo se centra en el diseño de arquitecturas hardware para el cálculo de ZNCC tanto en el dominio espacial como en el espectral. Con la notación utilizada para el cálculo de ZNCC, ecuación 4.3, serán necesarios el cálculo de CC, S y SS . Aunque todos ellos requieren un alto coste computacional, S y SS pueden ser optimizados tal y como se explica en el apartado 4.5. Teniendo esto en consideración, CC conllevaría el mayor coste computacional a la hora de calcular ZNCC y por tanto será su cálculo el que requerirá arquitecturas hardware específicas que consigan acelerarlo. A continuación se proponen arquitecturas hardware para el cálculo de CC eficientes tanto en consumo de recursos como en tiempo de ejecución (apartados 4.3 y 4.4).

4.3- Arquitectura espacial

La arquitectura espacial se basa en la implementación hardware de la ecuación 4.4 [Lindoso_07_e]. Esta ecuación requiere un gran número de multiplicaciones y acumulaciones, es decir operaciones MAC (“Multiply ACcumulation”). En un microprocesador estas operaciones son las que incrementan el coste computacional al calcular la correlación cruzada, ya que no se pueden ejecutar de una forma paralela. Sin embargo las operaciones MAC se pueden implementar en hardware de una manera altamente eficiente con un multiplicador, un sumador y unos cuantos registros.

Las nuevas generaciones de FPGAs contienen un gran número de multiplicadores empotrados que pueden ser utilizados para implementar operaciones MAC. Este tipo de operaciones se requieren para multitud de cálculos, destacando su amplia utilización en el campo del procesado digital de la señal, y la mayoría de los fabricantes de FPGAs han incluido en muchas de sus familias estructuras complejas para la implementación eficiente de operaciones MAC. En particular, el fabricante Xilinx denomina a estas estructuras DSP slices [Xil_DSPV4].

En la Figura 4.1 se muestra la arquitectura del DSP slice de Xilinx para la familia Virtex4 [Xil_V4]. Tal y como se aprecia en la figura, tiene dos entradas de datos A y B con un ancho de bus de 18 bits y una entrada de acarreo C con un ancho de bus de 48 bits. Dependiendo del tipo de conexión realizada entre los bloques DSP slices, pueden utilizarse salidas/entradas de alta velocidad (PCOUT, BCOUT, PCIN, BCIN). Estas salidas/entradas sirven para interconectar DSP slices entre sí y tienen un ancho de bus de 48 bits. Esta arquitectura posibilita la realización de un amplio conjunto de operaciones con los datos de entrada dando lugar a MAC de alta velocidad. Los bloques DSP slice deben configurarse para realizar la operación deseada dentro del conjunto de operaciones posibles.

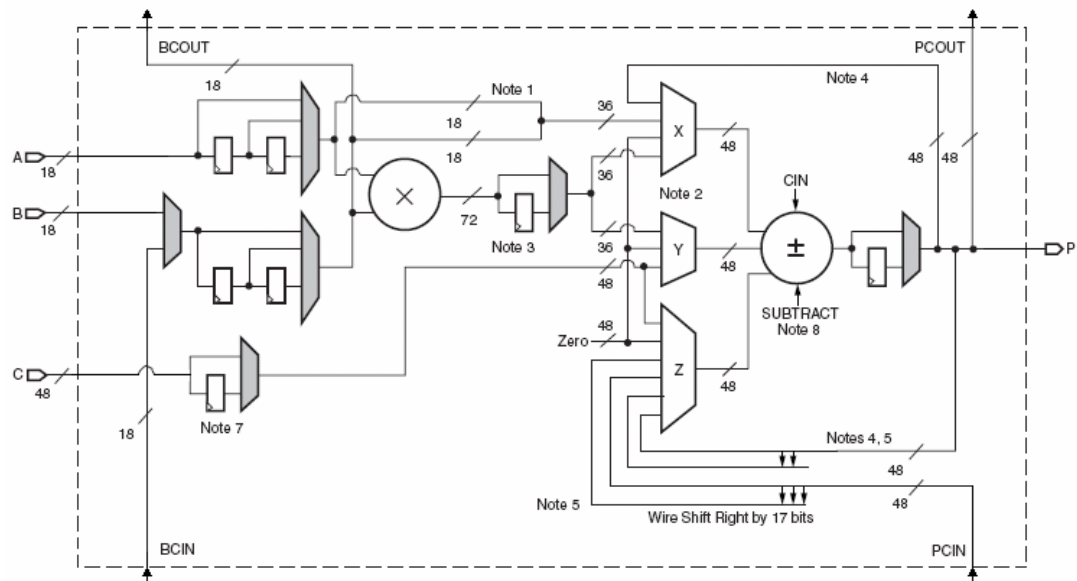


Figura 4.1. Bloque DSP Slice de Xilinx para la familia Virtex-4

La Figura 4.2 muestra la arquitectura espacial diseñada para la implementación hardware de la correlación cruzada. Esta arquitectura está basada en el concepto de ventana deslizante (“sliding window”) y utiliza como bloque de diseño básico los DSP slices descritos anteriormente. La arquitectura está compuesta por los siguientes elementos:

- Matriz de cálculo de correlaciones

Contiene una matriz de DSP slices. Cada fila de la matriz realiza el cálculo de la correlación cruzada de una fila de la imagen patrón (T) con una fila de la imagen de entrada (I).

Cada DSP slice de la matriz de correlaciones realiza una operación MAC y pasa el resultado al siguiente DSP slice de esa misma fila. Esto quiere decir que la matriz de correlaciones está organizada como una matriz sistólica en la que cada fila calcula la correlación de una fila de las imágenes de entrada (I) y patrón (T).

La imagen T se almacena en los registros de entrada de los DSP slices. Si el patrón tuviera un tamaño mayor que el número de DSP slices disponible dentro de la FPGA, podría ser almacenado en una memoria auxiliar realizada con bloques de RAM dentro de la FPGA. Para realizar el cálculo habría que subdividir el patrón en partes, cuyo tamaño sería dependiente del número de DSP slices disponibles, y repetir el cálculo para cada una de las partes. En el proceso habrá que actualizar los registros de entrada de los DSP slices cada vez que se finalice el cálculo de una de las partes.

- Líneas de retardo

Se utilizan para pasar los resultados intermedios de una fila de la matriz de correlaciones a la siguiente fila.

- Memoria de entrada

Almacena la imagen de entrada (I). Se implementa típicamente con bloques de RAM de la FPGA. No es necesario utilizar memoria externa salvo que la imagen sea demasiado grande.

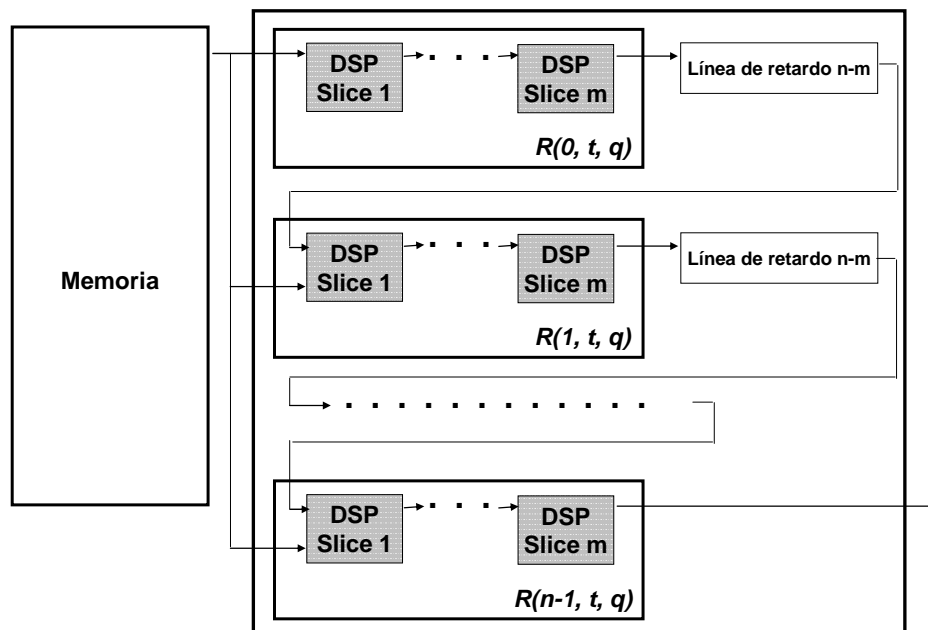


Figura 4.2. Arquitectura espacial para el cálculo de la correlación cruzada

En la Figura 4.3 se muestra como se realiza la interconexión de los DSP slices pertenecientes a una misma fila. En la Figura 4.3 se ha simplificado el esquema del DSP slice de la Figura 4.1, mostrando únicamente la configuración del DSP slice que se utiliza en este diseño. La Figura 4.3 muestra que en cada fila existen dos entradas de datos (A y B) y una línea de acarreo (P) que sirve para acumular los resultados parciales y poder pasar las sumas parciales a la siguiente fila mediante la línea de retardo, tal y como muestra la Figura 4.2.

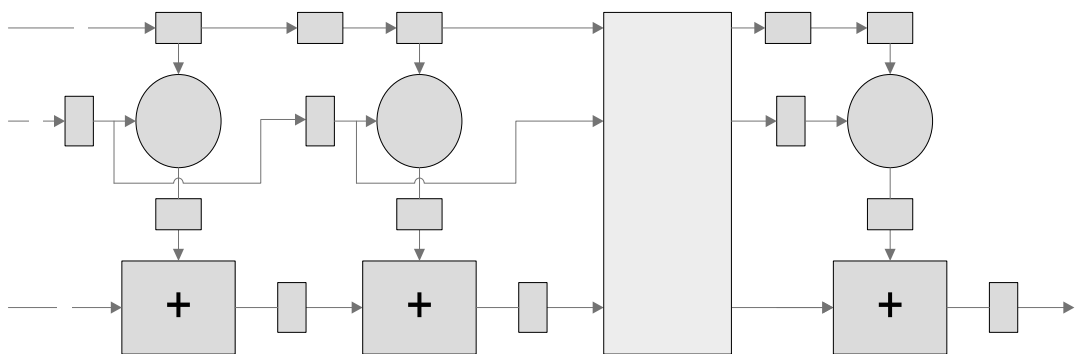


Figura 4.3. Interconexión de DSP slices dentro de una misma fila

Para minimizar el tiempo de cálculo y mejorar la eficiencia de la arquitectura todas las filas de la matriz de correlaciones toman los mismos datos de la memoria de entrada, imagen I, y datos diferentes del patrón provenientes de los registros de entrada de los DSP slices. Con esta estructura, en cada ciclo de reloj la memoria de entrada suministra un dato que pasa al primer DSP slice de cada fila de la matriz de correlaciones. Los resultados de las operaciones MAC y los datos de entrada provenientes de la memoria van pasando, dentro de la misma fila de la matriz de correlaciones, de un DSP slice al siguiente. Este proceso se realizará hasta que se complete una fila de la matriz de entrada, en ese momento será cuando se obtenga en el último DSP slice de cada fila de la matriz de correlaciones el resultado de la correlación cruzada de una fila de I con una fila de T.

El cálculo de la correlación cruzada necesita la suma de n correlaciones de fila consecutivas obtenidas en diferentes instantes de tiempo. Esta suma se realiza mediante la acumulación del resultado obtenido para cada correlación de fila y el

paso del mismo a la siguiente fila. Para realizar este proceso es necesario insertar una línea de retardo de $n-m$ ciclos de reloj entre la salida del último DSP slice de una fila y el primero de la siguiente, siendo n y m el tamaño de las filas de I y T respectivamente. Para obtener una implementación eficiente las líneas de retardo están formadas por LUTs (“Look-Up Table”) empleados como registros de desplazamiento. Con ello se evita malgastar DSP slices para realizar la acumulación de resultados parciales.

De manera más formal, la correlación se define recursivamente de la siguiente manera:

$$R(i, p, q) = R(i-1, p, q) + \sum_{j=0}^{m-1} T(i, j) I(p+i, q+j) \quad (4.10)$$

En la ecuación 4.10 $R(i, p, q)$ representa la correlación de las i primeras filas para un desplazamiento de (p, q) píxeles. Se considera que $R(-1, p, q) = 0$ y $CC(p, q) = R(n-1, p, q)$. Realizando el cambio de variable $t = i + p$ se llega a la siguiente expresión:

$$R(i, t, q) = R(i-1, t-1, q) + \sum_{j=0}^{m-1} T(i, j) I(t, q+j) \quad (4.11)$$

La ecuación 4.11 se implementa en la matriz de los DSP slices tal y como se detalla a continuación. Para cada ciclo de reloj un nuevo píxel $I(t, q)$ se lee de la memoria de entrada. El índice t se incrementa cuando el índice q ha completado una fila de la imagen. El DSP slice de la posición (i, j) tiene $T(i, j)$ en uno de sus registros de entrada y opera con un determinado valor de I que se pasará al siguiente DSP slice de esa fila. El paso del valor de I entre los componentes de la misma fila se realiza cada ciclo de reloj, ya que cada ciclo de reloj se lee un nuevo dato de la memoria de entrada.

Teniendo esto en cuenta, para un determinado ciclo de reloj y dados t y q , cada fila de DSP slices está calculando $R(i, t, q)$ para un diferente valor de $i = 0, \dots, n-1$. Esto quiere decir que en el mismo instante de tiempo se están calculando a la vez diferentes correlaciones de fila.

El resultado de la correlación de la fila anterior, $R(i-1, t-1, q)$, se suma al principio de cada fila. Para llevarlo a cabo es necesario introducir la línea de retardo ya que $R(i-1, t-1, q)$ se obtiene en el cálculo de la fila anterior, es decir en un instante de tiempo anterior.

Esta arquitectura permite el cálculo de $CC(p, q)$ para $p, q \in [0, \dots, n-1]$ en un solo paso de memoria, si el tamaño del patrón es igual que el de la matriz de DSP slices. Cada correlación requiere el mismo número de operaciones MAC que el tamaño del patrón, pero la arquitectura descrita puede realizar en un solo ciclo de reloj un número de operaciones MAC equivalentes al tamaño de la matriz de DSP slices. Teniendo esto en cuenta, el número de ciclos de reloj necesarios para realizar todas las correlaciones de fila es equivalente al tamaño de la imagen de entrada más la latencia de la correlación de una fila. Normalmente la latencia de la correlación de una fila es despreciable.

Si el tamaño del patrón es mayor que el número de DSP slices disponibles, es necesario dividir en bloques el patrón para realizar el cálculo por partes y después sumar los resultados.

El número total de ciclos de reloj necesarios para realizar CC puede estimarse mediante la siguiente ecuación:

$$N_{clk} = \text{Tamaño}(I) \times \text{Tamaño}(T) / \text{Tamaño}(\text{Matriz DSP}) \quad (4.12)$$

En esta arquitectura la velocidad se ve limitada por la memoria de entrada y las líneas de retardo. Aún así, cabe destacar que las FPGAs contienen pequeños bloques de RAM empotrados que son bastante rápidos [Xil_BRAM]. Por ejemplo la familia Virtex 4 de Xilinx [Xil_V4] contiene bloques de RAM de 18 Kbit que tienen una frecuencia nominal de 500 MHz. Sin embargo la frecuencia disminuye a

medida que es necesario construir bloques de memoria grandes, debido al multiplexado de los datos, retardando con ello el proceso de lectura/escritura de la memoria.

Para realizar un bloque de mayor tamaño sin tener penalización en el tiempo de acceso se ha optado por utilizar una frecuencia de reloj dos veces más lenta y doblar el ancho del bus de datos. Esto quiere decir que cada posición de memoria almacena dos datos que corresponden a dos píxeles consecutivos. Con esto se equilibra la penalización en el tiempo de acceso a la memoria, ya que en cada acceso se obtendrán dos datos.

En las líneas de retardo también se ha utilizado esta arquitectura para el bus de datos. Las líneas de retardo están compuestas por registros de desplazamiento que se implementan con celdas lógicas, que son más lentas que los DSP slices. Si se dobla el ancho del bus de datos de las líneas de retardo se puede trabajar a la mitad de frecuencia para proporcionar datos a un ritmo eficiente a los DSP slices.

4.4- Arquitectura espectral

La arquitectura espectral se basa en la implementación hardware de la ecuación 4.9 [Lindoso_07_e]. En este caso se realiza el cálculo de la correlación cruzada en el dominio de Fourier. Para la implementación de la ecuación 4.9 son necesarias una multiplicación compleja conjugada y 3 transformadas de Fourier bidimensionales, dos directas y una inversa. Normalmente se puede considerar que la transformada de Fourier del patrón se realiza previamente al cálculo de la correlación cruzada y por lo tanto solo serán necesarias dos transformadas de Fourier, una directa y otra inversa.

La arquitectura espectral se muestra en la Figura 4.4. En esta arquitectura las transformadas de Fourier se implementan mediante FFT (*“Fast Fourier Transform”*) bidimensionales, en las que las imágenes se rellenan con ceros si su

tamaño no es potencia de dos.

Para esta arquitectura las imágenes se dividen en filas o columnas para calcular en paralelo tantas FFTs como sea posible. Estos bloques de cálculo de FFT se han denominado FFT slices. El número máximo de FFT slices depende del tamaño de las imágenes y de la FPGA utilizadas.

Según muestra la Figura 4.4, se utilizan 3 memorias para esta arquitectura: M1 para el almacenamiento de la imagen de entrada y el resultado, M_T para el almacenamiento de la FFT del patrón, $F(T)$, y M2 para el almacenamiento de resultados intermedios. Todas estas memorias tienen el mismo tamaño, el doble del tamaño de la imagen de entrada, ya que el resultado de la FFT bidimensional es una matriz de números complejos. Todos los bloques de memoria se implementan con bloques de memoria de la FPGA con lo cual no es necesario utilizar memorias externas.

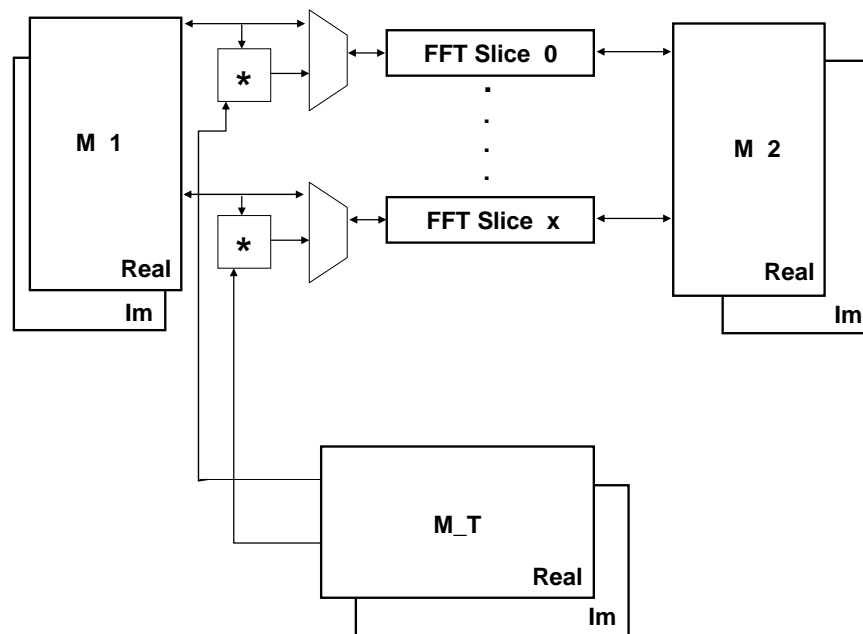


Figura 4.4. Arquitectura espectral para el cálculo de la correlación cruzada

El cálculo de las FFT bidimensionales se realiza mediante el siguiente proceso:

1. Cálculo de la FFT de las filas de M1 y almacenamiento del resultado en M2.
2. Cálculo de la FFT de las columnas de M2 y almacenamiento del resultado en M1. Al finalizar este paso M1 contiene $F(I)$.
3. Cálculo de la FFT inversa de las filas de M1 multiplicadas por el complejo conjugado de M_T , que contiene $F(T)$, y almacenamiento del resultado en M2.
4. Cálculo de la FFT inversa de las columnas de M2 y almacenamiento del resultado en M1.

Idealmente este proceso debería ser llevado a cabo por tantos FFT slices como filas tenga la imagen de entrada. En la práctica esto no es posible debido a la gran cantidad de recursos que consume cada FFT slice. Por tanto cada paso del proceso se realizará por un número de FFT slices que será inferior al número de filas de la imagen de entrada, debiendo repetirse el número de veces necesario para alcanzar el número total de filas.

El máximo número de FFT slices que puede contener una determinada FPGA depende del tamaño de las imágenes y del número de LUTs, DSP slices y bloques de memoria RAM que tenga disponible la FPGA elegida. Cualquiera de estos elementos puede ser el que limite el máximo número de FFT slices.

Para que el acceso a las memorias sea lo más eficiente posible, M1 y M2 se han dividido en bloques de manera que proporcionen suficientes puertos entrada/salida a cada FFT slice. Estos bloques contienen un subconjunto de filas en M1 y de columnas en M2. También se ha separado mediante bloques independientes la parte real de la imaginaria en todas las memorias.

Cada FFT slice necesita leer y escribir en multitud de bloques de memoria, requiriendo, además de lógica adicional que controle los accesos, algún mecanismo

que evite colisiones en los accesos a los bloques. Para evitar colisiones se han utilizado retardos en el funcionamiento de los FFT slices. En concreto se ha retardado cada FFT slice respecto al FFT slice anterior un ciclo de reloj durante los pasos 1 y 3 del proceso expuesto anteriormente. Gracias a estos retardos el acceso a las memorias se realiza de forma diagonal, en lugar de realizarse de forma horizontal o vertical.

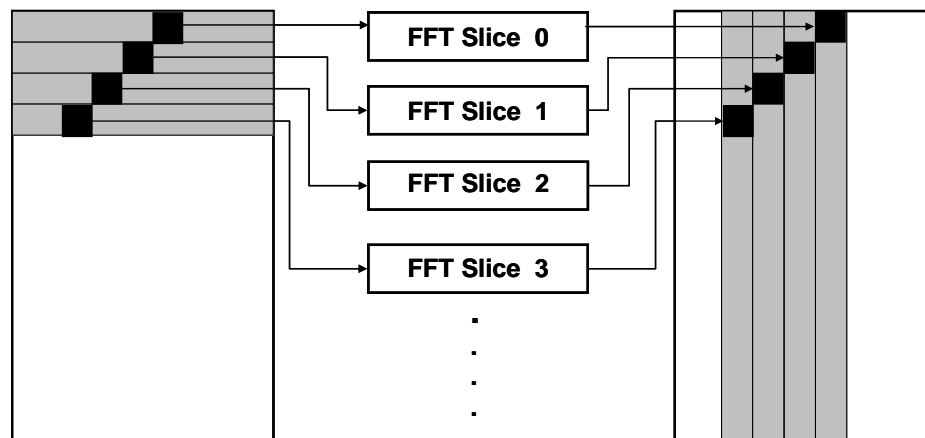


Figura 4.5. Organización del acceso a las memorias

Tal y como puede apreciarse en la Figura 4.5, mediante estos accesos diagonales es posible evitar colisiones, ya que en cualquier instante de tiempo cada FFT slice accede a una fila o columna diferente, y por tanto puede leer y escribir datos en todos los ciclos de reloj.

4.5- Normalización y comparación

Para completar el desarrollo de la ecuación 4.3, correlación cruzada normalizada con media cero ZNCC, es necesario incluir arquitecturas para la suma de píxeles de la imagen de entrada, S , ecuación 4.5, y la suma de cuadrados de los píxeles de la imagen de entrada, SS , ecuación 4.6.

$$S(p, q) = \sum_{i=0}^{i=n-1} \sum_{j=0}^{j=m-1} I(p+i, q+j) \quad (4.5)$$

$$SS(p, q) = \sum_{i=0}^{i=n-1} \sum_{j=0}^{j=m-1} I(p+i, q+j)^2 \quad (4.6)$$

S y SS pueden ser calculadas de la misma forma en la que se realiza el cálculo de CC(p, q) descrito en los apartados 4.3 y 4.4, ya que S es equivalente a realizar CC de I con la matriz identidad y SS es equivalente a realizar la correlación cruzada de I consigo mismo. La realización de estos cálculos aprovechando la misma arquitectura implica un retardo asociado, ya que para completar ZNCC es necesario recorrer las imágenes 3 veces (una por cada operación: CC, S y SS), mientras que con una arquitectura dedicada, el cálculo de CC, S y SS se puede realizar en paralelo, aprovechando el suministro de datos proveniente de la memoria cada ciclo de reloj.

En particular, S(p, q) puede calcularse mediante las siguientes ecuaciones recursivas [Lindoso_05]:

$$RA(p, q) = RA(p, q-1) + I(p, q) \quad (4.13)$$

$$RD(p, q) = RA(p, q+m-1) - RA(p, q-1) \quad (4.14)$$

$$CA(p, q) = CA(p-1, q) + RD(p, q) \quad (4.15)$$

$$S(p, q) = CA(p+n-1, q) - CA(p-1, q) \quad (4.16)$$

En las ecuaciones 4.13 - 4.16, m es la dimensión horizontal del patrón y n la dimensión vertical del patrón T, y todos los elementos con índices -1 se consideran de valor cero.

El cálculo de S y SS se realiza incluyendo desplazamientos. Normalmente la imagen I es de mayor tamaño que el patrón T y éste se desplaza dentro de la imagen

realizando correlaciones parciales. Para ello es necesario obtener S y SS de forma parcial sobre las ventanas de I coincidentes con la posición del patrón desplazado. El cálculo descrito en las ecuaciones 4.13 - 4.16 se realiza de la siguiente manera: RA acumula la suma de todos los elementos de una fila de la imagen I y RD resta de la acumulación realizada por RA la zona que no está dentro de la ventana considerada, es decir RD acumula la suma de los píxeles que se encuentran dentro de la ventana de correlación en la dirección horizontal. Esta ventana de correlación tiene el tamaño del patrón. Una operación similar se realiza por columnas con CA y S para conseguir que el resultado final contenga todos los elementos de la ventana de correlación y con ello se obtiene como resultado la suma de los píxeles, S .

Considerando las ecuaciones 4.13 – 4.16 se puede calcular $S(p, q)$ en un solo paso de memoria utilizando 2 sumadores y dos restadores, que es equivalente a utilizar 4 DSP slices, y algunas líneas de retardo. $SS(p, q)$ se puede calcular de la misma forma añadiendo un multiplicador al principio del proceso para obtener $I^2(i, j)$. Un esquema de la arquitectura resultante se muestra en la Figura 4.6.

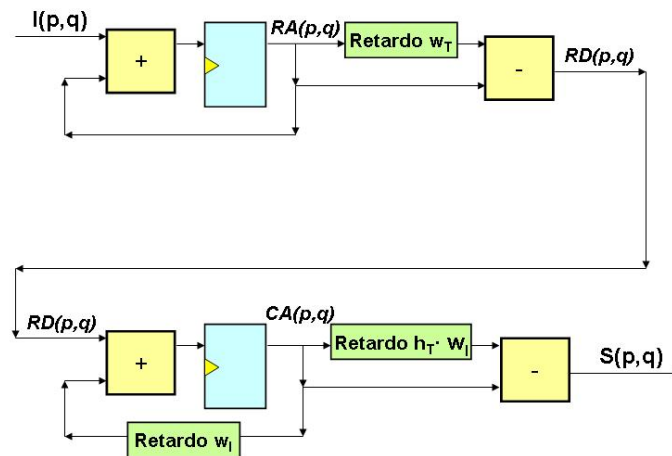


Figura 4.6. Arquitectura para el cálculo de $S(p, q)$

En la Figura 4.6, w_T es la dimensión horizontal del patrón (número de elementos de una fila), h_T es la dimensión vertical del patrón (número de elementos de una columna) y w_I es la dimensión horizontal de la imagen de entrada.

Gracias a esta aproximación, el cálculo de $S(p, q)$ y $SS(p, q)$ se puede sincronizar con el cálculo de $CC(p, q)$, con lo cual no se necesitaría tiempo adicional para la normalización.

A su vez también es necesario encontrar el máximo de $ZNCC(p, q)$, utilizando para ello la ecuación 4.8. Para implementar esta ecuación, sólo son necesarios 4 DSP slices para precisión simple (multiplicadores de 18x18 bits de la FPGA). Para ello se utilizan los cuadrados del numerador y denominador de $ZNCC(p1, q1)$ que se almacenan en cada iteración y se comparan con los actuales. Para obtener mayor precisión se pueden utilizar DSP slices adicionales. La búsqueda del máximo se puede realizar mientras se están calculando $CC(p, q)$, $S(p, q)$ y $SS(p, q)$ con lo cual el incremento en el tiempo de ejecución se manifestaría mediante un pequeño incremento de la latencia.

4.6- Resultados experimentales

En este apartado se presentan los resultados experimentales obtenidos para las arquitecturas descritas en los apartados anteriores de este capítulo. Se han considerado implementaciones software y hardware de la arquitectura espacial y la espectral para el cálculo de CC. A lo largo de este apartado la arquitectura espacial de la correlación cruzada se denominará CC Espacial y la arquitectura espectral de la correlación cruzada se denominará CC Espectral.

En el apartado 4.6.1 se explicarán decisiones relativas a la implementación de las arquitecturas así como el criterio para la selección de la FPGA en la que se ha llevado a cabo la implementación. El apartado 4.6.2 presenta los resultados obtenidos en cuanto a consumo de recursos y rendimiento de las arquitecturas

hardware y la aceleración conseguida respecto a implementaciones software. La aplicación de estas arquitecturas a la comparación de huellas dactilares mediante técnicas de correlación se analiza en el apartado 4.6.3.

4.6.1- Implementación de las arquitecturas y selección de FPGA

Aunque la mayoría de las familias de FPGAs podrían utilizarse para la implementación de las arquitecturas hardware CC espacial y CC espectral, se ha realizado un análisis preliminar para determinar qué familia es la que mejor se adapta a las características de dichas arquitecturas.

En primer lugar cabe destacar que el tamaño de la imagen es relevante para ambas arquitecturas en cuanto a rendimiento y en consumo de recursos, tanto en implementaciones software como en hardware. En todos los experimentos realizados se han utilizado imágenes de 256x256 píxeles.

La arquitectura CC Espacial requiere una gran cantidad de operaciones MAC. Debido a esto, la FPGA empleada para su implementación debe tener un gran número de DSP slices. La familia SX de la FPGA Virtex 4 de Xilinx [Xil_V4] presenta una gran cantidad de DSP slices ya que se diseñó para procesamiento digital de señal. En esta familia cada DSP slice puede realizar multitud de variaciones de operaciones MAC a muy alta velocidad alcanzando una frecuencia máxima de funcionamiento de 500 MHz. Dentro la familia Virtex-4 SX [Xil_V4] la FPGA más grande es la XC4SX55 con 512 DSP slices y es la que se ha utilizado para realizar los experimentos de la arquitectura CC espacial que se exponen en el apartado 4.6.

Cabe destacar que en la actualidad ya está disponible la tecnología Virtex-5 [Xil_V5], que proporciona mayores prestaciones y mayor volumen de recursos hardware, en particular de DSP slices. Los resultados que se presentan a continuación están realizados con Virtex-4 [Xil_V4], pero son extrapolables a

Virtex-5, presumiblemente con un aumento de las prestaciones.

En el caso de la implementación de CC Espectral, la eficiencia aumentará con el número de FFT slices que se ubiquen en la FPGA. Para la implementación de los FFT slices de la arquitectura CC espectral se ha utilizado el core de Xilinx *Fast Fourier Transform Core* [Xil_FFT]. Este core tiene tres posibles arquitecturas: *Radix 2 minimum resources*, *Radix 4 Burst I/O* y *Pipeline Streaming I/O*. La utilización de una u otra depende de la capacidad de la FPGA utilizada y de los requisitos temporales que se impongan al cálculo. Evaluando las tres posibles arquitecturas, la más rápida es la arquitectura *Pipeline Streaming I/O* con el menor retardo en el suministro de resultados (el retardo es equivalente al tamaño de los datos de entrada) pero con la mayor ocupación de área. De las tres arquitecturas, el mejor compromiso tamaño/velocidad lo presenta la arquitectura *Radix 4 Burst I/O*, que utiliza aproximadamente la mitad del área que la arquitectura *Pipeline* y tiene una penalización en velocidad que depende del número de puntos de la FFT. En cuanto a la ocupación, para una FFT de 256 puntos la arquitectura *Radix 4* necesita 9 DSP slices.

En los experimentos realizados para la arquitectura CC espectral se han utilizado FFT de 256 puntos ya que se han considerado imágenes de 256x256 píxeles. Para esta arquitectura también se ha elegido la misma FPGA, XC4SX55 [Xil_V4] con 512 DSP slices. La elección de esta FPGA se ha basado en el consumo de recursos de los FFT slices que necesitan gran cantidad de bloques de RAM y DSP slices. Ya que ninguna subfamilia de Virtex-4 presenta un alto número de estos dos recursos en el mismo dispositivo, se optó por la subfamilia SX que presenta mayor cantidad de ambos recursos en conjunto. La otra posibilidad era utilizar la subfamilia FX con gran número de bloques de RAM, pero en este caso los DSP slices que contiene la FPGA son tan escasos que el número de FFT slices en paralelo desciende considerablemente en comparación con la subfamilia SX.

Considerando FFT slices de 256 puntos y la FPGA XC4SX55, el número máximo de FFT slices que caben en el dispositivo son 16.

4.6.2- Consumo de recursos y rendimiento

Los resultados obtenidos están resumidos en las Tablas 4.1 y 4.2. Los resultados de ambas tablas se han obtenido con la herramienta ISE de Xilinx para imágenes de 256x256 píxeles y con la FPGA XC4SX55-11, la mayor FPGA de la subfamilia SX de la Virtex-4 de Xilinx.

Para los experimentos de la arquitectura CC espacial se han utilizado 3 patrones de tamaños diferentes: 12x12, 16x16 y 20x20. Todos ellos caben dentro de la FPGA sin necesidad de hacer el cálculo por bloques. La arquitectura CC espectral no depende del tamaño del patrón mientras sea menor que la imagen de entrada. En este caso los resultados dependen del número de FFTs, es decir FFT slices, que se puedan realizar en paralelo, que es 16 para la FPGA utilizada.

Implementación	Slices	Bloques RAM	DSP slices
CC Espacial 12x12	2731 (11%)	52 (16%)	147 (28%)
CC Espacial 16x16	3903 (15%)	60 (18%)	259 (50%)
CC Espacial 20x20	4653 (18%)	68 (21%)	403 (78%)
CC Espectral (16 FFT slices)	19612 (79%)	304 (95%)	208 (41%)
Normalización y Comparación (16x16) CC espacial	930 (3%)	16 (5%)	12 (2%)

Tabla 4.1. Resultados Experimentales (consumo de recursos de la FPGA)

La Tabla 4.1 muestra el consumo de recursos de la FPGA para las arquitecturas descritas en los apartados 4.3, 4.4 y 4.5. Los recursos evaluados son número de slices, bloques de RAM y DSP slices.

La Tabla 4.2 muestra el rendimiento alcanzado por las diversas arquitecturas y su aceleración respecto a implementaciones software.

Implementación	SW T_{CC} (ms)	HW T_{clk} (ns)	HW T_{CC} (ms)	Aceleración SW CC Espectral	Aceleración SW CC Espacial
CC Espacial 12x12	62	2.477	0.162	240	383
CC Espacial 16x16	110	2.438	0.160	244	688
CC Espacial 20x20	172	2.490	0.163	239	1055
CC Espectral (16 FFT slices)	39	4.069	0.145	270	760

Tabla 4.2. Resultados experimentales (rendimiento)

La segunda columna de la Tabla 4.2 muestra el tiempo necesario para completar el cálculo de la correlación cruzada para implementaciones SW, (SW Tcc). Estos resultados se obtuvieron con una librería propia compilada habilitando las opciones de optimización temporal. Resultados similares pueden obtenerse con otras librerías como por ejemplo [opencv]. Los resultados obtenidos con las implementaciones software se obtuvieron con un PC Pentium IV a 3 GHz con 1 GB de memoria. Las dos columnas siguientes de la Tabla 4.2 muestran el periodo mínimo de reloj (HW Tclk) y el tiempo necesario para realizar el cálculo de la correlación cruzada de dos imágenes (HW Tcc) para las implementaciones de las arquitecturas hardware. Las dos últimas columnas de la Tabla 4.2 muestran la aceleración de las implementaciones hardware respecto a las implementaciones software. Las tres

primeras filas de las tablas 4.1 y 4.2 corresponden a las implementaciones de la arquitectura CC espacial para patrones de tamaño 12x12, 16x16 y 20x20 respectivamente.

Considerando la implementación de la arquitectura CC espacial, su mayor limitación es el número de DSP slices disponibles en la FPGA utilizada. Los patrones de gran tamaño, como por ejemplo de 32x32, se deben dividir en bloques para realizar los cálculos. Esto implica que el tiempo de procesamiento se incrementa de manera proporcional al número de bloques que se deben procesar. Cabe destacar que aunque el patrón tuviera que dividirse porque necesite más DSP slices de los disponibles, su almacenamiento se haría en los bloques de memoria interna de la FPGA ya que hay recursos suficientes para ello. Para incluir la normalización y comparación para determinar el máximo de ZNCC es necesario añadir 12 DSP slices y algunos bloques de RAM (16 bloques para un patrón de 16x16, 20 bloques para un patrón de 20x20, y así sucesivamente) y slices para implementar las líneas de retardo. Como se explicó en el apartado 4.5, la normalización y comparación pueden calcularse a la vez que la correlación cruzada sin decrementar el rendimiento del sistema.

La cuarta fila de las tablas 4.1 y 4.2 muestra los resultados para la implementación de la arquitectura hardware CC espectral con 16 FFT slices. Esta arquitectura consume una gran cantidad de los tres recursos evaluados en las tablas 4.1 y 4.2, slices, bloques de RAM y DSP slices. Sin embargo, como se aprecia en los resultados de la tabla 4.2, esta arquitectura solo mejora las prestaciones muy ligeramente respecto a la implementación hardware de la arquitectura CC espacial.

La mejora que se consigue con la arquitectura espectral es menor de la esperada por diversos motivos. El primero de ellos es que cada FFT slice consume una gran cantidad de recursos, limitando con ello el número de FFT slices que pueden procesar datos en paralelo. Además de esto, el periodo de reloj mínimo es mayor que el de la implementación de la arquitectura CC espacial porque los retardos se encuentran en la lógica en lugar de en los DSP slices o en la memoria. A pesar de estos inconvenientes esta arquitectura puede ser ventajosa para tamaños de patrón

grandes, ya que para esta arquitectura el tiempo de procesado es independiente del tamaño del patrón.

La arquitectura CC espectral presenta desventajas adicionales: la imagen utilizada debe ser potencia de 2 y el resultado puede verse afectado por errores debidos al redondeo. Además de esto, si la imagen debe ser comparada con diversos patrones, o por ejemplo varias rotaciones del mismo patrón, se necesita una gran cantidad de memoria, ya que se debe almacenar la FFT de cada patrón.

Las implementaciones de las arquitecturas hardware muestran una aceleración de más de dos órdenes de magnitud con respecto a las implementaciones software, como se muestra en las dos últimas columnas de la Tabla 4.2. Si se toma como ejemplo la implementación de la arquitectura CC espacial para un patrón de 16x16, el cálculo de la correlación cruzada es 688 veces más rápido que una implementación software espacial y 244 veces más rápido que una implementación software espectral.

Por otra parte, si se considera la normalización estas cifras de aceleración se multiplican aproximadamente por tres, ya que para las arquitecturas hardware no se necesita tiempo adicional para el cálculo de S y SS.

4.6.3- Aplicación a la comparación de huellas dactilares.

En este apartado se exponen los resultados conseguidos con las arquitecturas descritas en los apartados 4.3 – 4.5 para los algoritmos de comparación de huellas dactilares basados en técnicas de correlación. Estos algoritmos utilizan el cálculo de la correlación de manera local, y considerando traslaciones y rotaciones, lo que ocasiona una elevada carga computacional.

Las huellas dactilares utilizadas en los experimentos pertenecen a la base de datos del FVC 2000 2A [FVC2000].

Los experimentos se han llevado a cabo considerando 13 rotaciones: 6 en sentido de las agujas del reloj, 6 en sentido contrario y la imagen sin aplicarle rotación.

El tamaño de las regiones locales consideradas varía entre 12x12 píxeles y 50x50 píxeles. Estos tamaños se han determinado empíricamente para las huellas de la base de datos utilizada. Se han buscado zonas lo suficientemente pequeñas para evitar la distorsión pero lo suficientemente grandes para que contuvieran información distintiva.

Para asegurar que una cantidad suficiente de la imagen se considera en la comparación, el número de regiones locales varía de manera inversamente proporcional al tamaño de las mismas. Esto quiere decir que si se utilizan regiones más pequeñas para evitar la distorsión implicará que un número mayor de regiones debe ser correlado para obtener un nivel aceptable de precisión en la comparación. En estos experimentos, se utilizan 54 regiones para un tamaño de 12x12 píxeles y únicamente 3 regiones si el tamaño es de 50x50 píxeles.

En el caso de la arquitectura CC espacial, si la región es mayor que el número de DSP slices disponibles en la FPGA es necesario dividir el cálculo en bloques tal y como se describe en el apartado 4.2. A pesar de esto, la imagen y las divisiones de la misma pueden almacenarse dentro de la FPGA sin necesidad de utilizar memoria externa. En este caso, el cálculo de la correlación se realiza de la siguiente manera. El primer bloque se cargará en los registros de entrada de la matriz de correlación de los DSP slices. La carga de los datos se realiza en serie y tarda un número de ciclos equivalente al número de DSP slices que contenga la matriz de correlación. Una vez finalizado el cálculo de la correlación de ese bloque, el siguiente bloque se cargará en los registros de entrada de los DSP slices para repetir el cálculo. El proceso se repetirá tantas veces como el número bloques en los que se haya dividido el patrón/imagen.

La Tabla 4.3 muestra el rendimiento de las implementaciones software de un algoritmo completo de comparación de huellas dactilares basado en técnicas de

4. ACCELERACIÓN HARDWARE DEL CÁLCULO DE LA CORRELACIÓN

correlación. Todas las medidas han sido obtenidas con PC Pentium IV a 3 GHz y 1GByte de memoria. Para las dos arquitecturas hardware, CC espacial y CC espectral, se han utilizado tamaños de regiones de 12x12, 16x16, 20x20, 32x32 y 50x50 píxeles.

Algoritmo de Correlación	Tiempo por Región (ms)	# Regiones	Tiempo Total (s)
CC Espacial 12x12	328.85	54	17.76
CC Espacial 16x16	432.52	27	11.68
CC Espacial 20x20	787.98	18	14.18
CC Espacial 32x32	1988.63	9	17.90
CC Espacial 50x50	3373.06	3	10.12
CC Espacial 12x12	764.32	54	41.27
CC Espacial 16X16	764.32	27	20.64
CC Espectral 20X20	764.32	18	13.76
CC Espectral 32X32	764.32	9	6.88
CC Espectral 50X50	764.32	3	2.29

Tabla 4.3. Rendimiento de las implementaciones software de la comparación de huellas dactilares basada en técnicas de correlación

La segunda columna de la Tabla 4.3 muestra el tiempo necesario para realizar el cálculo de la correlación de una región de una imagen y todas las posibles regiones del mismo tamaño de la otra imagen (la fase de preprocesado no ha sido incluida en las medidas de tiempo). La tercera columna muestra el número de correlaciones de región necesarias para realizar la comparación de dos huellas. El tiempo total para realizar el algoritmo de comparación se muestra en la cuarta columna. Estos resultados se han obtenido con imágenes de 256x256 píxeles.

En los algoritmos de comparación de huellas dactilares basados en técnicas de correlación, el tiempo de comparación de dos huellas es invariable para un tamaño de bloque fijo. Por lo tanto el tiempo requerido para comparar una huella con toda una base de datos será el tiempo necesario para realizar una comparación (columna 3 de la Tabla 4.3, dependiendo de la arquitectura y el tamaño del bloque elegidos) multiplicado por el número de huellas que contenga la base de datos.

Según la Tabla 4.3 el tiempo necesario para realizar los cálculos con implementaciones software de CC espacial varía entre 10 y 17 segundos y entre 2 y 41 segundos para implementaciones software de CC espectral. Como se aprecia en estos resultados, CC espectral muestra un mejor rendimiento para tamaños superiores a 20x20. Aún así, los resultados obtenidos no son aceptables para ninguna aplicación que requiera respuesta en tiempo real.

La Tabla 4.4 compara el tiempo total necesario para realizar la comparación en implementaciones hardware y software considerando en ambos casos las implementaciones CC espacial y espectral para diferentes tamaños, similarmente a la Tabla 4.3. La segunda columna de la Tabla 4.4 muestra el tiempo total para realizar la comparación para las implementaciones software y la tercera columna muestra ese mismo tiempo para las implementaciones hardware. La cuarta columna muestra la aceleración que se consigue al utilizar las arquitecturas hardware.

Evaluando los resultados, la arquitectura hardware CC espacial consigue un aumento en la aceleración a medida que el tamaño es más pequeño que el número de DSP slices disponible. La aceleración disminuye si el tamaño es mayor que el

4. ACCELERACIÓN HARDWARE DEL CÁLCULO DE LA CORRELACIÓN

número de DSP slices disponibles debido a la partición de la imagen en bloques. En estos casos el rendimiento mejora si se considera la arquitectura hardware CC espectral.

Algoritmo de Correlación	Tiempo SW (s)	Tiempo HW (s)	Aceleración
CC Espacial 12x12	17.76	0.114	156
CC Espacial 16x16	11.68	0.056	208
CC Espacial 20x20	14.18	0.038	372
CC Espacial 32x32	17.90	0.057	315
CC Espacial 50x50	10.12	0.032	321
CC Espectral 12x12	41.27	0.153	270
CC Espectral 16X16	20.64	0.076	270
CC Espectral 20X20	13.76	0.051	270
CC Espectral 32X32	6.88	0.025	270
CC Espectral 50X50	2.29	0.008	270

Tabla 4.4. Rendimiento de implementaciones hardware de la comparación de huellas dactilares basada en técnicas de correlación

Las implementaciones hardware requieren un tiempo adicional para la transferencia de datos a la FPGA. Si se considera un bus PCI, se necesitarían menos de 2 milisegundos para la transferencia de una imagen de 256x256 píxeles y menos de 4 milisegundos para la transferencia de la transformada de Fourier de una imagen de 256x256 píxeles. Este tiempo es aceptable para la arquitectura CC espacial. Además en este caso se podrían enviar patrones en paralelo con la ejecución del cálculo de la correlación, ya que su almacenamiento no afecta a la ejecución del cálculo.

Sin embargo, para la arquitectura CC espectral la transferencia de la transformada de Fourier de cada región conlleva mucho más tiempo. En este caso es mejor transferir a la FPGA el patrón y realizar su FFT dentro de la FPGA. Los resultados obtenidos en la Tabla 4.4 han utilizado este modelo de transferencia.

La Figura 4.7 muestra gráficamente la variación del tiempo de procesamiento para CC espacial y espectral en las implementaciones hardware y software. El eje X representa el tamaño de la región, variando entre 12 (12x12 píxeles) y 50 (50x50 píxeles), y el eje Y representa en escala logarítmica el tiempo medio de procesamiento en milisegundos para una única región.

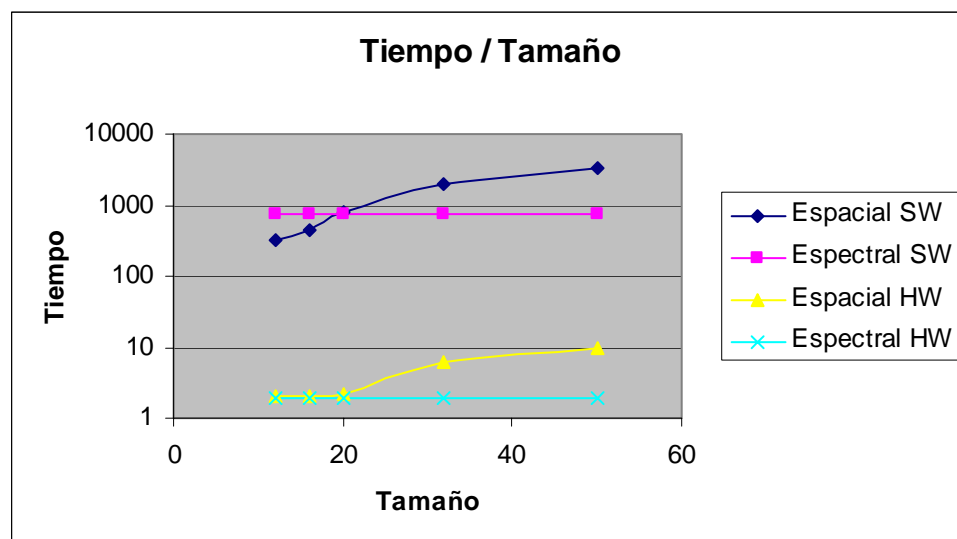


Figura 4.7. Tiempo de procesamiento versus tamaño de las regiones

Los resultados obtenidos muestran que los algoritmos de comparación de huellas dactilares basados en técnicas de correlación que utilizan CC espacial software consumen típicamente más de 10 segundos en realizar una comparación exhaustiva. Este tiempo puede disminuir alrededor de 2 segundos si se considera CC espectral software con un número pequeño de regiones grandes. Esta mejora del rendimiento se debe a que la FFT no aumenta la carga computacional de manera proporcional al aumento del tamaño pero sí que lo hace de manera proporcional al número de regiones empleadas. Estos tiempos en ningún caso son aceptables para la mayoría de las aplicaciones a menos que se lleven a cabo diversas simplificaciones en los cálculos para reducir la carga computacional a costa de reducir con ello la precisión de la comparación.

Evaluando los resultados de las implementaciones hardware, en el caso peor, la comparación puede realizarse en alrededor de 100 milisegundos. Este tiempo disminuye considerablemente considerando un tamaño típico de las regiones. Por lo tanto el tiempo de procesamiento puede reducirse en dos órdenes de magnitud, pasando de segundos a milisegundos y consiguiendo que este tipo de algoritmos sea viable para una amplia gama de aplicaciones, incluso aquellas que requieren respuesta en tiempo real.

Aunque estos resultados se han obtenido con una FPGA de altas prestaciones, es posible escalar las arquitecturas para FPGAs de bajo coste sin reducir excesivamente el rendimiento. Actualmente existen familias de bajo coste, como la Spartan-3A DSP [Xil_Spartan3], que también contienen grandes cantidades de DSP slices (hasta 126) a una velocidad nominal de 250 MHz.

Los resultados mostrados en las Tablas 4.3-4.4 y en la Figura 4.7 utilizan un algoritmo de correlación genérico que realiza la correlación de un patrón de tamaño variable (desde 12x12 hasta 50x50 píxeles) sobre una imagen de entrada de tamaño fijo de 256x256. En los cálculos están incluidos todos los posibles desplazamientos del patrón sobre la imagen de entrada incluyendo 13 rotaciones. En estos cálculos se han utilizado diferente número de regiones en función del tamaño del patrón considerado. Para comparar los resultados obtenidos con los algoritmos expuestos en el capítulo 3 es necesario tener en cuenta que los experimentos realizados no

coinciden en los tamaños de las imágenes, ni en el número de regiones a comparar, resultando por tanto en un número diferente de correlaciones a realizar. A continuación se detallarán los parámetros de los dos algoritmos expuestos en el capítulo 3 para poder realizar una comparación de rendimiento hardware/software.

El algoritmo de baja resolución del capítulo 3 utiliza los siguientes parámetros para realizar la comparación:

- Tamaño del patrón: 50x50 píxeles.
- Tamaño de la imagen de entrada: 100x100 píxeles, eliminando un borde de 17 píxeles en ambas direcciones resultando un área efectiva de 83x83 píxeles.
- Número de rotaciones: 13.
- Número de regiones a comparar: 3.

El algoritmo de alta resolución utiliza los siguientes parámetros:

- Tamaño del patrón: 25x25 píxeles.
- Tamaño de la imagen de entrada: 150x150 píxeles, eliminando un borde de 25 píxeles en ambas direcciones resultando un área efectiva de 125x125 píxeles.
- Número de rotaciones: 13.
- Número de regiones a comparar: 4.

En base a los parámetros utilizados en estos algoritmos a continuación se muestra en la Tabla 4.5 el rendimiento comparado de los algoritmos tanto en software como en hardware y las aceleraciones conseguidas en cada caso.

En la Tabla 4.5 se ha utilizado la arquitectura espacial adaptada a los parámetros de cada uno de los algoritmos. Para el tiempo de las implementaciones software sólo se ha considerado la etapa de comparación de ambos algoritmos. Ambas

implementaciones software se han realizado en C, con un PC Pentium IV a 3 GHz y 1GByte de memoria.

Algoritmo	Tiempo SW (s)	Tiempo HW (s)	Aceleración
Baja Resolución	0.9	0.00336	268
Alta Resolución	1.350	0.00331	407

Tabla 4.5. Rendimiento de implementaciones hardware de los algoritmos de baja y alta resolución

Como se aprecia en la Tabla 4.5, los tiempos necesarios tanto para las implementaciones hardware como las software son menores que los de las Tablas 4.3 y 4.4. Esto es debido al menor número de correlaciones utilizado por los algoritmos que los detallados en dichas tablas. Particularmente influye considerablemente el tamaño de la imagen de entrada, ya que sobre ella se realizan los desplazamientos, que pasa de 256x256 a 83x83 píxeles en el caso del algoritmo de baja resolución, y a 125x125 píxeles en el caso del algoritmo de alta resolución.

La Tabla 4.5 confirma los resultados aportados por las tablas 4.3 - 4.4 que consideran algoritmos genéricos con alta carga computacional, obteniendo aceleraciones dos órdenes de magnitud en ambos casos.

4.7- Conclusiones

El cálculo de la correlación cruzada es un cálculo ampliamente utilizado en diversos campos de aplicación relacionados con el procesamiento de imágenes. Sin embargo su uso se ve limitado en la mayoría de las aplicaciones debido a la gran complejidad computacional que conlleva. En este capítulo se han presentado dos arquitecturas hardware que aceleran el cálculo de la correlación cruzada tanto en el dominio espacial como en el espectral. Utilizando las arquitecturas hardware se puede reducir el tiempo requerido para el cálculo de CC en dos órdenes de magnitud sin pérdida de precisión con respecto a implementaciones software en un PC de altas prestaciones.

La mejora del rendimiento obtenida por la arquitectura espectral es menor de lo esperado inicialmente. Esto es debido a la gran cantidad de recursos que consume esta arquitectura. En consecuencia, el número de FFT slices que se pueden implementar a la vez se ve drásticamente reducido, reduciendo con ello la aceleración. Además, en este caso, el cálculo que se realiza, FFT bidimensional, es mucho más complejo que el realizado en el caso de la arquitectura espacial, aumentando el tiempo de cálculo. Por otro lado también hay que tener en cuenta que los resultados de la arquitectura espectral pueden verse afectados por el redondeo y además se necesita una mayor cantidad de memoria si se consideran varios patrones o rotaciones de un mismo patrón. A pesar de todo esto, para patrones de gran tamaño es mejor utilizar esta arquitectura ya que, a diferencia de la arquitectura espacial, para una imagen de tamaño fijo, un aumento en el tamaño del patrón no implica un aumento de la carga computacional.

La aceleración conseguida gracias a las arquitecturas hardware permite utilizar las técnicas de correlación en aplicaciones en tiempo real. En particular, las arquitecturas descritas en este capítulo se han aplicado a un algoritmo de comparación de huellas dactilares basado en técnicas de correlación. Los resultados

experimentales demuestran que la comparación de huellas dactilares podría pasar de segundos a milisegundos gracias a las arquitecturas hardware descritas en este capítulo. Como consecuencia de esto se podrían realizar correlaciones de una manera más amplia sin perjudicar con ello los requisitos de rendimiento del sistema.

Con el objetivo de obtener un alto rendimiento se ha utilizado para las implementaciones de las arquitecturas hardware una FPGA Virtex-4 de la familia SX de Xilinx. Sin embargo las arquitecturas hardware descritas en este capítulo son escalables y pueden adaptarse fácilmente a las necesidades del sistema que se desee diseñar. Las arquitecturas podrían adaptarse para su implementación en FPGAs de bajo coste y aún así se podrían seguir obteniendo resultados en tiempo real. En todo caso, la adaptación requerirá un compromiso entre el rendimiento deseado y los recursos disponibles en la FPGA.

Capítulo 5:

ARQUITECTURAS HARDWARE PARA LA COMPARACIÓN EFICIENTE DE HUELLAS DACTILARES

5.1-	INTRODUCCIÓN.....	130
5.2-	DESCRIPCIÓN DE LA ARQUITECTURA DEL SISTEMA.....	132
5.3-	COMPONENTES DEL SISTEMA.....	134
5.3.1-	<i>FPGA</i>	134
5.3.2-	<i>Microprocesador</i>	135
5.3.3-	<i>Comunicaciones</i>	136
5.4-	COPROCESADOR HARDWARE	139
5.4.1-	<i>Arquitectura estática</i>	140
5.4.2-	<i>Arquitectura dinámica</i>	142
5.5-	ALGORITMO IMPLEMENTADO	147
5.5.1-	<i>Transformada wavelet</i>	148
5.5.2-	<i>ZNCC</i>	149
5.6-	SOFTWARE.....	151
5.7-	RESULTADOS EXPERIMENTALES	152
5.7.1-	<i>Precisión</i>	153
5.7.2-	<i>Consumo de recursos</i>	154
5.7.3-	<i>Rendimiento</i>	156
5.8-	CONCLUSIONES	162

5.1- Introducción

Una vez analizadas las características de los algoritmos de comparación de huellas dactilares basados en técnicas de correlación en el capítulo 3 y la aceleración hardware de los mismos en el capítulo 4, en este capítulo se estudian arquitecturas hardware a nivel de sistema que permitan una implementación eficiente de algoritmos de comparación de huellas dactilares. Estos algoritmos se enmarcan dentro del campo del procesamiento de imágenes y conllevan la realización de operaciones de complejidad computacional alta sobre un conjunto de datos de dimensiones elevadas. Tanto el trasiego de datos como la cantidad y complejidad de operaciones a realizar ralentizan las prestaciones de este tipo de algoritmos en implementaciones orientadas a un microprocesador convencional. Aunque el microprocesador utilizado sea de última generación, las operaciones complejas requieren muchos ciclos de reloj, tanto para la ejecución de las instrucciones como para la captura de los datos con los que se opera. Esta pérdida de prestaciones aumenta proporcionalmente al número de veces que se realiza la operación, siendo en este caso proporcional al tamaño de la imagen. Habitualmente se utilizan en este campo microprocesadores de alto rendimiento que llevan asociados un elevado consumo, gran complejidad del sistema y un aumento del coste que pueden ser inaceptables para muchas aplicaciones.

Una solución para acelerar operaciones concretas de alta complejidad es recurrir a coprocesadores especializados que se encargan de realizar ciertas tareas de manera optimizada. Sin embargo, las aplicaciones que conllevan operaciones con imágenes no se benefician en un aumento de prestaciones con este tipo de arquitecturas si no se proveen de una gestión eficiente del trasiego de datos y de la posibilidad de realizar la misma operación sobre un conjunto de datos a alta velocidad. Para el procesamiento de imágenes el aumento del rendimiento se produce cuando el sistema utilizado es capaz de realizar la misma operación para un conjunto de datos a alta velocidad, ya que en este caso la gestión eficiente de los datos es fundamental.

En este capítulo se estudiarán arquitecturas desarrolladas para SoPC (“System on Programmable Chip”). Los SoPC tienen como ventajas un reducido tamaño, bajo coste y capacidad intrínseca de reconfiguración y por tanto de adaptación a nuevas arquitecturas o actualizaciones. Además es factible pasar a una implementación SoC (“System on Chip”) si el volumen de la producción lo justifica.

En concreto, se proponen arquitecturas basadas en la utilización de un coprocesador hardware para aumentar las prestaciones de un microprocesador empotrado. En este caso el coprocesador no acelerará únicamente una instrucción del microprocesador si no que se buscará la paralelización de operaciones así como la gestión eficiente de los datos de entrada y salida. Las prestaciones de un microprocesador empotrado típico son muy inferiores a las de un microprocesador de gama alta como el que contiene un PC de última generación. Sin embargo, en este capítulo se demostrará que incluyendo el coprocesador apropiado se pueden conseguir prestaciones equivalentes.

Las arquitecturas para SoPC que se proponen están orientadas a la implementación de algoritmos verificación de huellas dactilares basados en técnicas de correlación. Estas arquitecturas buscan eficiencia en la gestión de datos así como en la realización de operaciones de alta complejidad para un conjunto de datos y operaciones características en el ámbito del procesado de imágenes. Los sistemas diseñados plantean una arquitectura de sistema eficiente formada por un coprocesador hardware unido a un microprocesador empotrado. En esta arquitectura el coprocesador se hará cargo de las tareas de alta complejidad computacional mientras que el microprocesador empotrado realizará el resto del algoritmo y controlará las comunicaciones del sistema.

El coprocesador hardware diseñado para el sistema está basado en la arquitectura espacial descrita en el apartado 4.3 del capítulo 4, presentando dos aproximaciones: coprocesador estático y coprocesador dinámico. La diferencia que existe entre ambas es la capacidad de reconfiguración en tiempo de ejecución del coprocesador dinámico. Esto hace posible el cambio de tamaño y de tarea a realizar por el coprocesador. Además, se ha optado por una reconfiguración de grano grueso que se realiza directamente a través del microprocesador sin necesidad de utilizar

los mecanismos internos de la FPGA (regeneración del bitstream y descarga del mismo), reduciendo considerablemente el tiempo de reconfiguración. Con esta arquitectura se consigue un coprocesador versátil y altamente eficiente para operaciones con imágenes.

En este capítulo se detalla la arquitectura en los apartados 5.2 y 5.3, y el coprocesador hardware en el apartado 5.4, con sus dos posibles variantes: estática y dinámica. En los apartados 5.5 y 5.6 se describe la implementación del algoritmo de reconocimiento de huellas dactilares en el SoPC descrito. Finalmente en el apartado 5.7 se presentan los resultados experimentales. Las conclusiones de este capítulo se resumen en el apartado 5.8.

5.2- Descripción de la arquitectura del sistema

Para el prototipado del sistema de reconocimiento mediante huella dactilar se ha utilizado un SoPC debido a las numerosas ventajas que presenta: altas prestaciones, bajo coste, posibilidad de reconfiguración, reducido tamaño, etc.

El sistema propuesto está formado por una FPGA que contendrá un microprocesador empotrado y un coprocesador hardware que realizará las tareas de alta complejidad computacional, pudiendo así aliviar la carga soportada por el microprocesador y aumentando la eficiencia del sistema.

El único condicionante que se impone al sistema es que las imágenes de las huellas estén almacenadas en memoria externa a la FPGA. Ya que se está diseñando un sistema de reconocimiento mediante huella dactilar, al menos una de las imágenes (el patrón o “template”) debe estar almacenada en una base de datos y ser enviada cuando se realice la comparación. Esta base de datos puede ser local (almacenamiento dentro de la propia placa) o remota (servidor externo).

La imagen de entrada (capturada con un sensor de huella dactilar) se enviará al

microprocesador empotrado para realizar la comparación con la pertinente imagen o imágenes de la base de datos. La conexión del sensor con el sistema está fuera del ámbito de esta tesis, pero se podría realizar mediante una conexión USB, PCI, Ethernet o cualquier otra conexión de suficiente velocidad.

Según todo lo expuesto, el sistema consta de un microprocesador empotrado conectado a un coprocesador hardware y a un banco de memoria externo a la FPGA que contiene las imágenes a comparar. El esquema del sistema se muestra en la Figura 5.1.

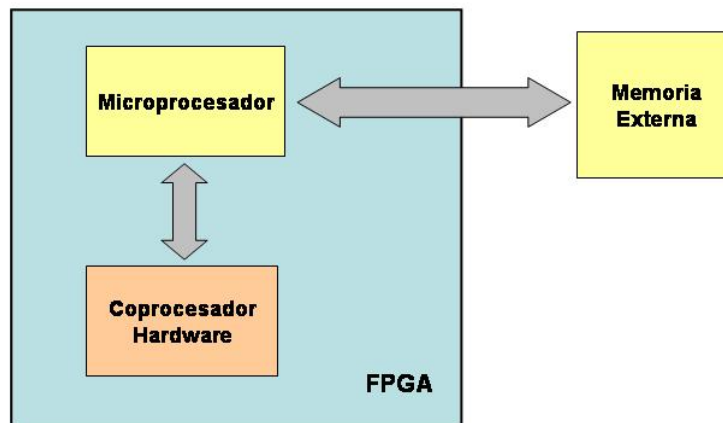


Figura 5.1. Arquitectura del sistema

Dentro del sistema descrito en la Figura 5.1 existen diversas opciones para la comunicación de los elementos del sistema. Estas opciones dependen en gran medida de la tecnología utilizada (FPGA y microprocesador) ya que cada fabricante presenta un abanico de opciones diferenciado, aunque normalmente son equivalentes. En el siguiente apartado, apartado 5.3, se detalla cada uno de los componentes elegidos para el sistema, encuadrados dentro de las opciones disponibles para sistemas empotrados que presenta el fabricante elegido para la implementación. Cabe destacar que aunque la implementación se haya diseñado para un fabricante en concreto, la arquitectura es portable a otros fabricantes.

5.3- Componentes del sistema

Para la implementación del prototipo del sistema se ha elegido el fabricante Xilinx [Xilinx]. En los siguientes subapartados se detallarán las características de los elementos del sistema, encuadrados dentro de las diversas opciones que ofrece el fabricante.

5.3.1- FPGA

Dentro del fabricante Xilinx se han buscado las familias que contengan DSP slices, necesarios para la arquitectura del coprocesador hardware, basado en la arquitectura directa descrita en el apartado 4.3 del capítulo 4. Las familias que presentan esa característica son Virtex-4 [Xil_V4] y Virtex-5 [Xil_V5]. Dentro de esas familias las que poseen mayor cantidad de DSP slices son las de la subfamilia SX. Concretamente, la implementación se ha prototipado sobre una placa de evaluación comercial, ML-506 [Xil_ML506] con una FPGA Virtex 5 SX: XC5VSX50TFFG1136, con 288 DSP slices.

5.3.2- Microprocesador

El fabricante Xilinx presenta 3 microprocesadores para los diseños SoPC: 2 microprocesadores “soft core”, PicoBlaze [Xil_pblaze] y MicroBlaze [Xil_mblaze], y un microprocesador “hard core”, PowerPC [Xil_PPC]. PicoBlaze es un microprocesador empotrado de 8 bits altamente optimizado para ocupar muy poco espacio en la FPGA, pero con prestaciones muy limitadas. MicroBlaze es un microprocesador empotrado de 32 bits con arquitectura Harvard con mayores prestaciones que PicoBlaze, pero también con un mayor consumo de área. PowerPC es un microprocesador RISC de 32 bits “hard core”, es decir, está empotrado en el silicio en fabricación y por tanto ese área no puede utilizarse como dispositivo reprogramable.

De entre las tres opciones disponibles se optó por los microprocesadores “soft core” ya que proporcionan una mayor flexibilidad al sistema, descartando por tanto el microprocesador Power PC. De entre las dos opciones restantes se escogió MicroBlaze ya que proporciona mejores prestaciones. El mayor consumo de área requerido por el microprocesador MicroBlaze no es un problema para el sistema a diseñar. En la Figura 5.2 se muestra un esquema de la arquitectura de MicroBlaze.

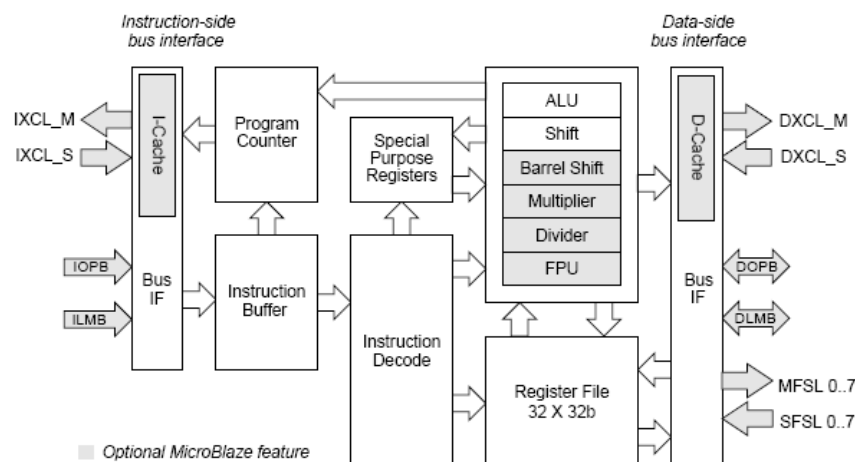


Figura 5.2. Arquitectura de MicroBlaze

5.3.3- Comunicaciones

Una vez elegido el microprocesador, existen tres posibilidades para la comunicación de los elementos del sistema dentro de la plataforma de Xilinx: el bus OPB (On-Chip Peripheral Bus) [Xil_OPB], el bus PLB (“Processor Local Bus”) [Xil_PLB] y el bus FSL (“Fast Simple Link”) [Xil_FSL]. Además de estos buses existe el bus LMB (“Local Memory Bus”) [Xil_LMB] que sirve para acceder a memoria local. Este último bus no se ha considerado ya que las imágenes no están almacenadas en memoria local, pero será utilizado para el almacenamiento de la memoria de programa (código de programa, variables y constantes).

A continuación se resumen las características más destacadas de cada uno de los buses:

➤ Bus FSL:

- Es un canal de comunicación punto a punto unidireccional. Se utiliza para envíos rápidos entre microprocesador empotrado y otros elementos de la FPGA.
- Su velocidad de transferencia es de 2 ciclos de reloj por dato. Escribe y lee directamente en los registros del microprocesador.
- Se pueden establecer hasta 8 conexiones FSL bidireccionales.
- Permite la utilización de FIFOs (“First In First Out”) para la comunicación. Se pueden implementar con LUTs (“Look-Up Table”) o BRAM (“Random Access Memory”) y dependiendo de la configuración su tamaño varía de 1-8K (32 bits).
- El control y el protocolo de comunicación lo debe establecer el usuario en función de las necesidades de la comunicación a realizar.
- El hardware necesario para implementar este bus es reducido y puede variar entre 3 y 62 LUTs.

➤ Bus PLB

- Bus síncrono de 64 bits.
- Protocolo simple que garantiza accesos en BRAM en un ciclo de reloj.
- Proporciona un interfaz bidireccional periférico-microprocesador.
- Interrupciones.
- Múltiples registros para lectura/escritura.
- FIFOs de lectura/escritura con soporte para modo empaquetado.
- Transferencia en modo ráfaga.
- Transferencia en modo DMA (“Direct Memory Access”).
- El hardware necesario para la implementación de este bus depende de la configuración utilizada pudiendo variar entre 135 y 4546 LUTs.

➤ Bus OPB:

- Interfaz bidireccional periférico-microprocesador de 32 bits.
- Interrupciones.
- Múltiples registros para lectura/escritura.
- FIFOs de lectura/escritura con soporte para modo empaquetado.
- Transferencia en modo ráfaga.
- Transferencia en modo DMA.

A la vista de las características expuestas, los buses OPB y PLB tienen características similares. La diferencia más destacable es que el OPB ocupa menor área que el PLB.

El bus FSL se suele utilizar para conexiones de alta velocidad periférico-microprocesador. Aunque el FSL permite FIFOs, no puede realizar transferencias

en modo DMA. Teniendo en cuenta que las imágenes están situadas en memoria externa, la utilización de este bus genera varios inconvenientes para el sistema a diseñar:

- Ya que el envío se hace a través de registro es necesario pasar las imágenes desde la memoria externa al microprocesador (memoria de programa) y enviarlas mediante el FSL al coprocesador.
- Es necesario crear un protocolo de comunicación adaptado a los datos y las características del sistema que se quiere diseñar.
- No se puede utilizar DMA y por tanto el microprocesador no puede realizar ninguna otra tarea mientras se realizan transmisiones o recepciones. Debido a las características del sistema, estos envíos serán numerosos y conllevarán el trasiego de una gran cantidad de datos.
- La plataforma de Xilinx no facilita la depuración de este bus en el sentido microprocesador-periférico lo que dificulta en gran medida su utilización.

Teniendo en cuenta todas las posibles opciones se ha elegido el bus OPB para la comunicación del microprocesador con los periféricos. El bus OPB es similar al bus PLB y el cambio de uno a otro no conllevaría problemas adicionales.

En principio, el bus FSL sería el indicado para comunicar el coprocesador hardware con el microprocesador pero no se ha utilizado por los inconvenientes expuestos anteriormente. Aunque este bus es muy rápido para envíos a un periférico, no tiene modo DMA, con lo cual para envíos considerablemente grandes y continuados el microprocesador se queda bloqueado leyendo o escribiendo registros sin posibilidad de realizar otra tarea. Otro inconveniente importante es el paso de la imagen de memoria externa a memoria local, ya que es necesario para cualquier transferencia leer las imágenes dos veces, aparte del incremento de memoria local que se experimenta. A pesar de estos inconvenientes sería posible utilizar este bus para la comunicación con el coprocesador pero el sistema seguiría necesitando o bien el PLB o bien el OPB para la comunicación del resto de los elementos. Cabe destacar que al utilizar el bus FSL es necesario crear el protocolo

de comunicación mientras en los otros dos buses la transferencia DMA ya está implementada. También es destacable que la depuración vía XMD (“Xilinx Microprocessor Debugger”) no contiene al bus FSL y por tanto solo puede realizarse mediante el control del contenido de los registros de Microblaze dejando sin posibilidad de depuración el envío de datos hacia el coprocesador.

5.4- Coprocesador Hardware

El coprocesador hardware es el elemento fundamental de la arquitectura propuesta. El coprocesador hardware del SoPC se encargará de realizar las tareas con mayor complejidad computacional del algoritmo de comparación de huellas dactilares. Gracias a su utilización el microprocesador empotrado se liberará de la realización de estas tareas mejorando el rendimiento global del sistema.

Para el diseño del coprocesador se ha partido de la arquitectura espacial descrita en el apartado 4.3 del capítulo 4. Ya que esta arquitectura se utilizará ahora como un coprocesador dentro de un SoPC, es necesario modificar dicha arquitectura para adaptarla a las características del sistema. En el SoPC no se dispone de todos los DSP slices que contiene la FPGA ya que, dependiendo de la configuración y de los periféricos empleados, estos pueden necesitarse para otros elementos del sistema. Es por tanto necesario encontrar una configuración con una relación óptima entre prestaciones y área ocupada.

En este apartado se describen dos posibles opciones para el coprocesador hardware, a las que nos referiremos respectivamente como arquitectura estática y dinámica respectivamente.

5.4.1- Arquitectura estática

El coprocesador hardware está basado en la arquitectura espacial descrita en el apartado 4.3 del capítulo 4. La arquitectura estática es una versión simplificada en aras de obtener un coprocesador básico, sencillo y con un reducido consumo de recursos. En este caso, en lugar de utilizar una matriz de DSP slices, se utilizará una única fila de DSP slices como coprocesador. Teniendo una única fila no son necesarias las líneas de retardo para pasar los resultados parciales de una fila a la siguiente de la matriz de DSP slices, aunque se requerirá con el microprocesador un esfuerzo adicional para la recomposición de resultados finales.

El tamaño de fila elegido es de 48 DSP slices. Este tamaño es el número de DSP slices que contienen las columnas de DSP slices para las familias de Xilinx Virtex 4 [Xil_V4] y Virtex 5 [Xil_V5]. Utilizando un tamaño de fila menor o igual al tamaño de la columna de DSP slices es posible realizar todas las conexiones a alta velocidad.

El coprocesador realiza las operaciones CC, S y SS necesarias para calcular ZNCC. Aunque se utiliza una única arquitectura estática para las tres operaciones, a continuación se subdividirá la arquitectura en dos para facilitar la explicación de las diferentes operaciones.

La Figura 5.3 muestra la arquitectura del coprocesador hardware para el cálculo de CC. Como muestra la Figura 5.3 la arquitectura del coprocesador consta de una fila de n DSP slices y memorias para el almacenamiento de resultados parciales. Un sistema de control se encarga de gestionar el flujo de datos de entrada/salida en el coprocesador así como los datos que se están procesando en cada instante por la fila de DSP slices. Ya que en este caso no se opera con una matriz de DSP slices, será necesario componer los resultados obtenidos por filas para dar lugar a un resultado matricial. Esta arquitectura genera resultados cada ciclo de reloj, pero no todos los resultados generados son válidos ya que la arquitectura presenta latencia. Por lo tanto es necesario que el bloque de control seleccione los datos válidos de

correlación y los almacene en la memoria intermedia. Para las operaciones SS y S la arquitectura del coprocesador hardware se muestra en la Figura 5.4.

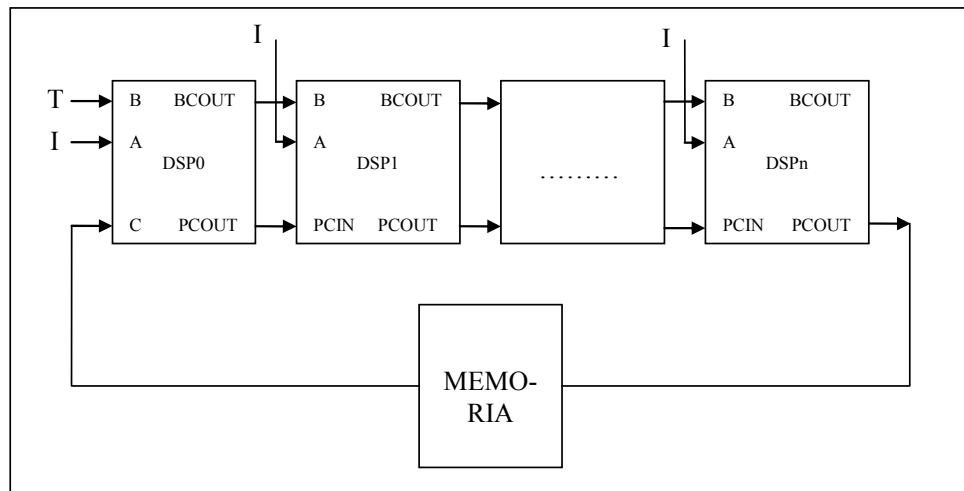


Figura 5.3. Coprocesador hardware (CC)

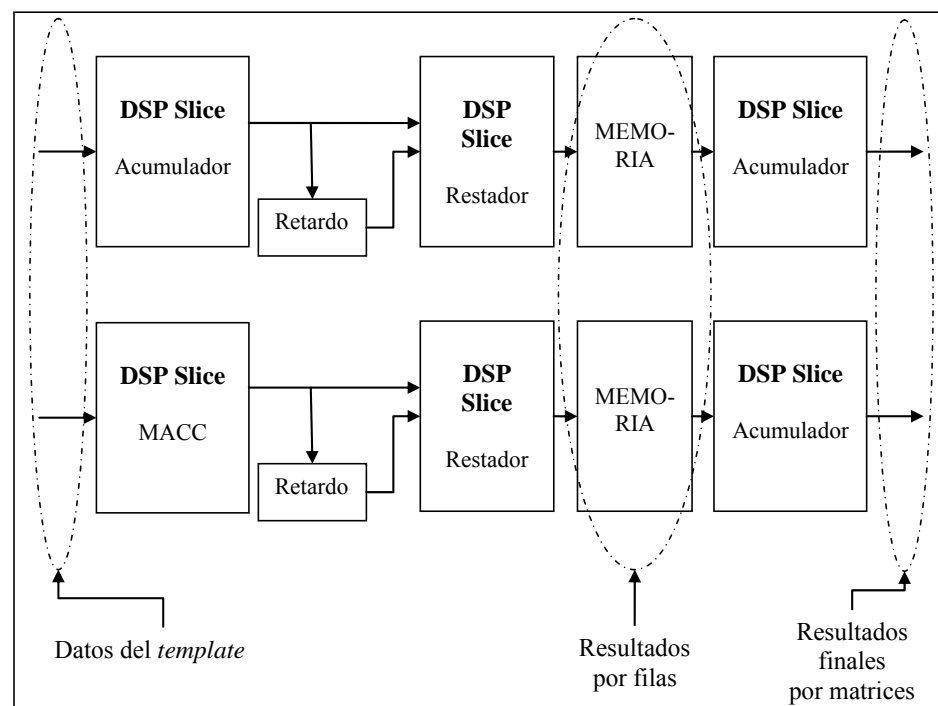


Figura 5.4. Coprocesador hardware (S y SS)

En este caso también es necesario el bloque de control para el manejo del flujo de datos entrada/salida y para la selección de resultados intermedios válidos. La arquitectura mostrada en la Figura 5.4 realiza la suma o suma de cuadrados por filas de una imagen. Para realizarlo se fundamenta en el esquema mostrado en la Figura 5.5. En este esquema se muestra que para ambas operaciones (S y SS) se realiza una suma con acumulación a la vez que una substracción con almacenamiento. Esta arquitectura es equivalente a la descrita en el apartado 4.5 del capítulo 4.

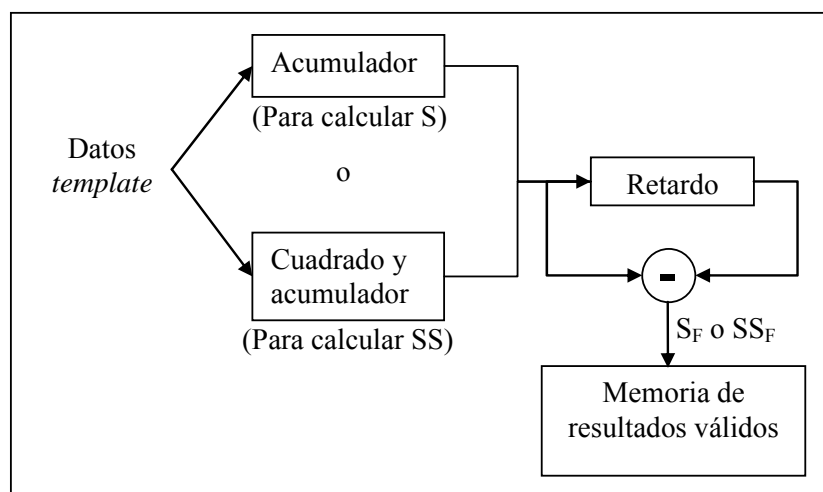


Figura 5.5. Esquema de cálculo de S y SS por filas

5.4.2- Arquitectura dinámica

La arquitectura dinámica plantea una solución más compleja y también más versátil con el objetivo de explotar más eficientemente los recursos ofrecidos por una FPGA dada [Lindoso_08]. El objetivo es que el coprocesador hardware sea capaz de realizar diferentes operaciones con diferentes tamaños de imágenes, de manera que pueda adaptarse fácilmente a diferentes tareas dentro de un rango de aplicaciones. Esto se consigue cambiando su arquitectura mediante reconfiguración de grano grueso realizada directamente por el microprocesador empujado. El

coprocesador propuesto es apto no sólo para reconocimiento de huellas dactilares sino para multitud de operaciones de procesamiento de imágenes.

Un aspecto crucial en el diseño de una arquitectura dinámica es su capacidad de reconfiguración. En principio, una FPGA proporciona una capacidad intrínseca de reconfiguración muy amplia, bien sea total o parcial, a través de su bitstream. Sin embargo, el tiempo de reconfiguración es proporcional al tamaño de dicho bitstream y por tanto puede ser muy largo. Es preciso, por tanto, encontrar un balance apropiado entre la reconfigurabilidad prevista y el tiempo de reconfiguración. Este balance se consigue limitando las opciones de reconfiguración a los aspectos estrictamente necesarios.

El coprocesador está orientado a realizar cálculos como la convolución, la correlación y el filtrado de imágenes. Todas estas operaciones requieren fundamentalmente una gran cantidad de operaciones de multiplicación-acumulación (MAC), que se implementan en las FPGAs con DSP slices [Xil_DSPV5]. Afortunadamente, la función de los DSP slices se puede reconfigurar dinámicamente sin necesidad de reprogramar la FPGA. Otros aspectos, como la interconexión entre los DSP slices u otros elementos del coprocesador, se pueden hacer también reconfigurables mediante un diseño apropiado. Los registros de configuración pueden hacerse accesibles al sistema de manera que puedan modificarse en tiempo de ejecución. En concreto, el coprocesador diseñado utiliza una matriz de DSP slices que se reconfigura directamente a través del microprocesador empujado.

La arquitectura del coprocesador se muestra en la Figura 5.6. Los elementos básicos de esta arquitectura son la matriz de DSP slices, el bloque de control y las FIFO de entrada/salida para el intercambio de datos a través del OPB con el microprocesador empujado.

El bloque de control es el que se encarga de la gestión de las FIFO de entrada/salida y el paso de datos desde las FIFOs al coprocesador y al microprocesador empujado. Este bloque también realiza la configuración de la matriz de DSP slices. Asimismo, el bloque de control gestiona los relojes de los

diferentes elementos dentro del coprocesador.

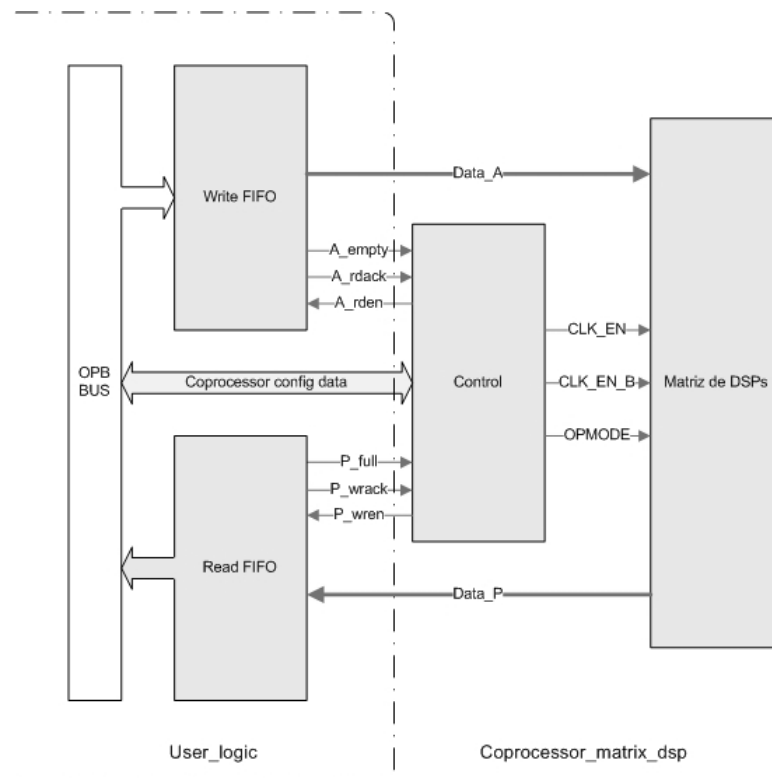


Figura 5.6. Arquitectura del Coprocesador

Dentro del coprocesador se utilizan dos conjuntos de datos que tienen dos relojes diferentes. Uno de estos conjuntos de datos contiene una imagen y el otro conjunto de datos contiene los coeficientes de un filtro o el patrón de la correlación. El segundo conjunto de datos necesita un reloj que se pare una vez se halla cargado el contenido adecuado en la matriz de los DSP slices.

El coprocesador contiene un conjunto de registros de configuración que se pueden programar desde el microprocesador empujado. Los registros de configuración son los siguientes:

- Operación del DSP slice (OPMODE, ALUMODE).
- Tamaño de la matriz de DSP slices.
- Tamaño de la imagen de entrada.

- Tasa de submuestreo. Gracias a este parámetro se permite el submuestreo de los resultados obtenidos con la matriz de DSP slices y su almacenamiento en la FIFO de salida.
- Operaciones de control, tales como el reset e inicio de la tarea del coprocesador.

Como ya se ha mencionado anteriormente el coprocesador se reconfigura directamente vía OPB sin necesidad de utilizar los mecanismos internos de reconfiguración proporcionados por la propia FPGA. Para optimizar el rendimiento del sistema los parámetros necesarios para la reconfiguración del coprocesador se han empaquetado en 4 transacciones del bus OPB. Una vez que la reconfiguración del coprocesador tiene lugar, el procesamiento de datos comienza en cuanto existan datos disponibles en la FIFO de entrada. Los resultados se almacenan en la FIFO de salida y el envío y recepción por parte del microprocesador empujado se realiza mediante DMA.

La Figura 5.7 muestra la arquitectura de la matriz de DSP slices. La matriz de DSP slices contiene DSP slices conectados en “pipeline”. El tamaño de las filas y el número de filas vienen determinados por la configuración que se realice del coprocesador. Para pasar los resultados de una fila de DSP slices a la siguiente son necesarias líneas de retardo. Esta arquitectura es totalmente equivalente a la descrita en el apartado 4.3 del capítulo 4, pero la operación realizada así como el tamaño de las filas de DSP slices y el número de filas son configurables.

Las dimensiones de la matriz están limitadas por el número de DSP slices disponibles en la FPGA utilizada y por su disposición dentro de la misma. Si se consideran las FPGAs Virtex 5 de Xilinx [Xil_V5] los DSP slices están agrupados en columnas de 48 DSP slices. Para alcanzar máximas prestaciones es necesario conectar DSP slices pertenecientes a la misma columna. Para la FPGA Virtex 5 SX50T [Xil_V5] el número máximo de columnas de DSP slices que puede utilizar el coprocesador es 5, ya que aunque la FPGA tiene 6 columnas, una de ellas la utiliza Microblaze.

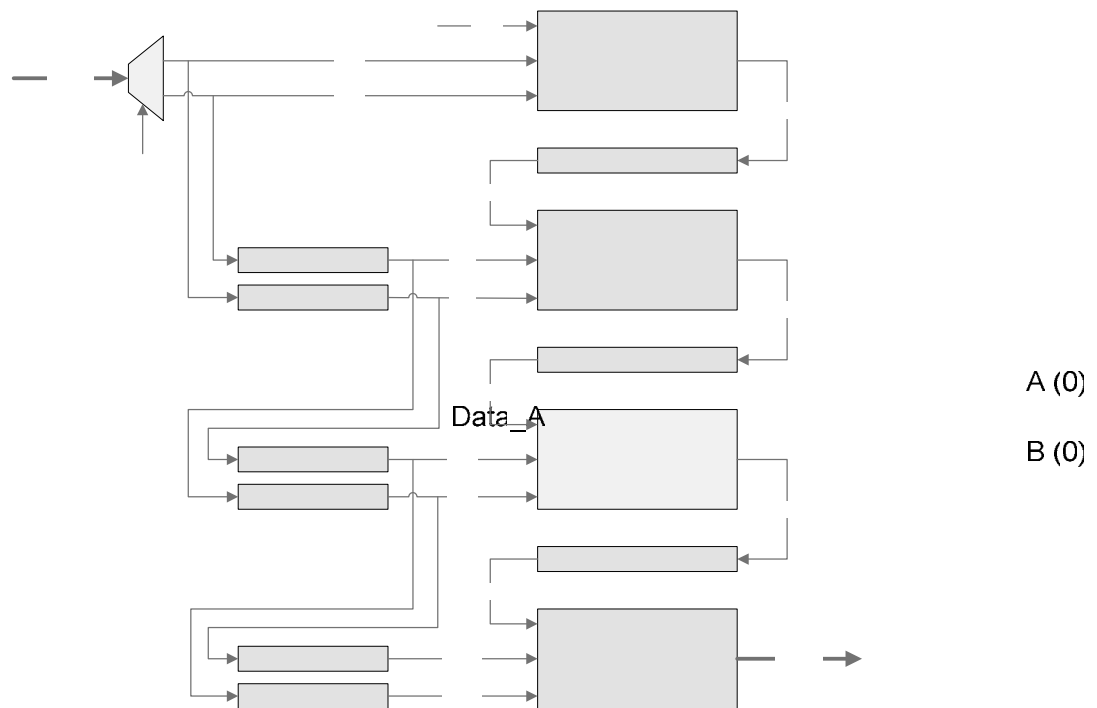


Figura 5.7. Arquitectura de la matriz de DSP slices

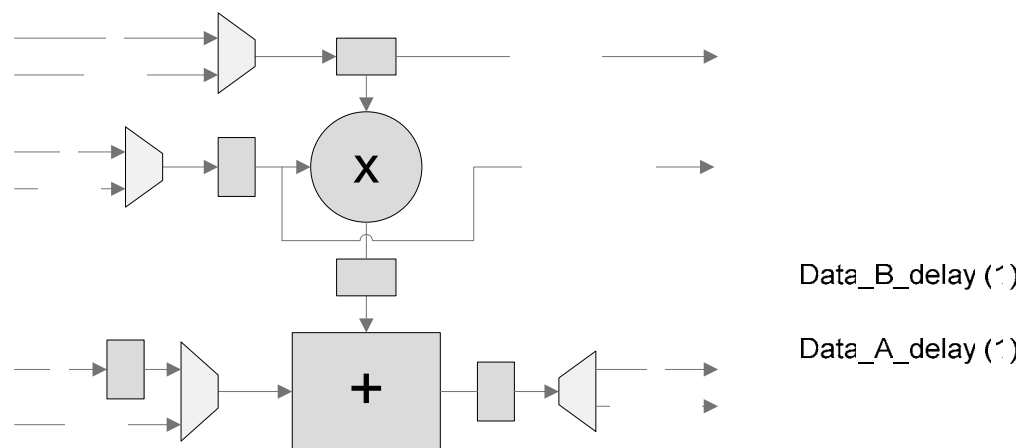


Figura 5.8. Arquitectura simplificada de un DSP Slice

Cada DSP slice [Xil_DSPV5] contiene un multiplicador de 24x18 y un acumulador de 48 bits. Una arquitectura simplificada de un DSP slice se muestra en la Figura 5.8. Los DSP slices tienen muchos más parámetros de configuración que los que se

utilizan para la reconfiguración vía OPB. Los parámetros restantes se han considerado estáticos para esta arquitectura.

Las Figuras 5.7 y 5.8 muestran que hay 3 rutas de datos dentro de la matriz de DSP slices: A, B y PCIN. A y B se utilizan para el paso de las imágenes de entrada, bien dos imágenes o bien una imagen y un conjunto de coeficientes. PCIN se utiliza para pasar los resultados de una fila de DSP slices a la siguiente. Como los datos tienen diferentes tamaños necesitan líneas de retardo de diferente tamaño (Data A delay, Data B delay y PCIN delay). El tamaño de las líneas de retardo también se establece en el proceso de reconfiguración del coprocesador.

5.5- Algoritmo implementado

El sistema descrito se ha utilizado para implementar el algoritmo de verificación de huellas dactilares para imágenes de alta resolución descrito en el capítulo 3, apartado 3.4. Esta aplicación dentro del campo del procesado de imágenes requiere respuesta en tiempo real y por lo tanto requiere altas prestaciones en rendimiento.

El algoritmo, tal y como se describe en el capítulo 3, tiene dos etapas: selección de regiones y comparación. Las tareas que conllevan mayor esfuerzo computacional son la transformada wavelet en la etapa de selección de regiones y ZNCC en la etapa de comparación. Para la implementación del algoritmo dentro del SoPC estas tareas con alta carga computacional se han realizado con el coprocesador y el resto del algoritmo lo realiza el microprocesador empujado. Para ello el coprocesador opera con cuatro modos de funcionamiento: transformada wavelet, suma de cuadrados (SS), suma (S) y correlación cruzada (CC). Las tres últimas operaciones son necesarias para el cálculo de ZNCC. La normalización y la composición de resultados a partir de los resultados parciales las realiza el microprocesador empujado. En los siguientes subapartados se describen detalladamente los diferentes modos de operación del coprocesador.

5.5.1- Transformada wavelet

La transformada wavelet [Mallat_98], [Gonzales_08] realiza un filtrado espacial y espectral que se lleva a cabo filtrando y submuestreando las imágenes de manera iterativa. El número de iteraciones depende del tamaño de la imagen y del nivel de la transformada wavelet.

Para el algoritmo utilizado se realizan transformadas wavelet diádicas de nivel 2 y la familia utilizada es Daubechies 2. Normalmente, al calcular la transformada wavelet, se realiza el filtrado de manera independiente para las direcciones vertical y horizontal. Sin embargo los filtros bidimensionales son más eficientes en las arquitecturas hardware. Por lo tanto se han utilizado 4 filtros bidimensionales en lugar de 8 filtros unidimensionales para cada nivel de la transformada wavelet. Teniendo en cuenta la familia utilizada, el tamaño de los filtros utilizados es 4x4. Los coeficientes de los filtros han sido escalados de manera que el coprocesador realice las operaciones eficientemente. El resultado es correctamente reescalado por el microprocesador empotrado.

Las cuatro matrices de coeficientes de los filtros se muestran a continuación:

$$\begin{aligned} HH &= \begin{bmatrix} 0.2333 & -0.4040 & 0.1083 & 0.0625 \\ -0.4040 & 0.6998 & -0.1875 & -0.1083 \\ 0.1803 & -0.1875 & 0.0502 & 0.0290 \\ 0.0625 & -0.1083 & 0.0290 & 0.0167 \end{bmatrix} & LH &= \begin{bmatrix} 0.0625 & -0.1083 & 0.0290 & 0.0167 \\ -0.1083 & 0.1875 & -0.0502 & -0.0290 \\ -0.4040 & 0.6998 & -0.1875 & -0.1083 \\ -0.2333 & 0.4040 & -0.1083 & -0.0625 \end{bmatrix} \\ HL &= \begin{bmatrix} 0.0625 & -0.1083 & -0.4040 & -0.2333 \\ -0.1083 & 0.1875 & 0.6998 & 0.4040 \\ 0.0290 & -0.0502 & -0.1875 & -0.1083 \\ 0.0167 & -0.0290 & -0.1083 & -0.0625 \end{bmatrix} & LL &= \begin{bmatrix} 0.0167 & -0.0290 & -0.1083 & -0.0625 \\ -0.0290 & 0.0502 & 0.1875 & 0.1083 \\ -0.1083 & 0.1875 & 0.6998 & 0.4040 \\ -0.0625 & 0.1083 & 0.4040 & 0.2333 \end{bmatrix} \end{aligned}$$

El modo de operación transformada wavelet del coprocesador utiliza una matriz de DSP slices de tamaño 4x4. Los coeficientes de los filtros y la imagen a transformar se envían al coprocesador vía OPB. El primer nivel de la transformada

wavelet diádica requiere realizar 4 filtrados bidimensionales diferentes sobre toda la imagen. Estos 4 filtros se han realizado de forma secuencial en el coprocesador. La arquitectura diseñada podría ser optimizada realizando varios filtros en paralelo. Con esta opción se podría aumentar el rendimiento del coprocesador pero la transferencia de datos se vería penalizada. Solamente se puede utilizar una FIFO para el envío con DMA del OPB, lo que dificulta la gestión y envío de los resultados de cuatro filtros en paralelo. Para gestionar los datos de salida de varios filtros en paralelo sería necesario realizar un almacenamiento temporal de los datos de salida, además de un sistema de control para realizar los envíos en el instante temporal adecuado a la única FIFO de salida.

En la arquitectura diseñada para el coprocesador reconfigurable, una vez que se obtienen los resultados, se almacenan en la FIFO de salida y se envían al microprocesador empujado vía OPB. Si se incluyeran elementos adicionales para el almacenamiento de resultados y su envío a través del OPB, la tasa de transferencia de datos se vería penalizada influyendo en el rendimiento del coprocesador así como en su capacidad de reconfiguración.

5.5.2- ZNCC

La correlación cruzada con media normalizada (ZNCC) se utiliza en este algoritmo para medir la similitud de las zonas de las huellas seleccionadas y poder determinar por tanto su grado de igualdad. Esta operación es comúnmente utilizada en el ámbito del procesamiento de imágenes para la comparación de patrones. Considerando las traslaciones de las imágenes, ZNCC se puede expresar mediante la siguiente fórmula:

$$ZNCC(p, q) = \frac{CC(T - \bar{T}, I(p, q) - \bar{I}(p, q))}{\|T - \bar{T}\| \cdot \|I(p, q) - \bar{I}(p, q)\|} \quad (5.1)$$

En la ecuación 5.1 T e I son las imágenes patrón e imagen de entrada respectivamente. \bar{T} e \bar{I} son las medias de esas imágenes y p y q son las traslaciones horizontal y vertical de la imagen de entrada.

Para acelerar el cálculo de ZNCC se han utilizado 3 modos diferentes de operación en el coprocesador que por consiguiente necesitará 3 reconfiguraciones para realizar ZNCC. Estos modos son: correlación cruzada (CC), suma (S) y suma de cuadrados (SS). La operación realizada por el coprocesador para cada modo se muestra en las siguientes ecuaciones:

$$CC(p, q) = \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} T(i, j) I(p+i, q+j) \quad (5.2)$$

$$S(p, q) = \sum_{i=0}^{i=n-1} \sum_{j=0}^{j=m-1} I(p+i, q+j) \quad (5.3)$$

$$SS(p, q) = \sum_{i=0}^{i=n-1} \sum_{j=0}^{j=m-1} I(p+i, q+j)^2 \quad (5.4)$$

Cada una de estas ecuaciones se realiza de manera independiente por el coprocesador. Cada uno de estos modos de operación se realiza con el máximo número de DSP slices disponibles en la matriz de DSP slices. El número de DSP slices disponibles depende de la FPGA utilizada. Por simplicidad se ha optado por utilizar matrices de DSP slices con dimensiones pares. Por ejemplo, para la FPGA Virtex 5 SX50T (XCVSX50T-1FFG1136) [Xil_V5], las dimensiones máximas de la matriz de DSP slices se han fijado en 4 filas de 48 DSP slices, requiriendo 192 DSP slices. También hay que tener en cuenta el tamaño de la imagen sobre la que se va a realizar la operación, pero éste normalmente es mayor que la matriz de DSP slices. Esto implica que el coprocesador, para todos estos modos, realiza la operación por partes y envía los resultados parciales al microprocesador empujado. El microprocesador es el encargado de generar el resultado final a partir de los parciales y de realizar la normalización de los resultados.

5.6- Software

A continuación se muestra un esquema (Figura 5.9) que resume el código ejecutado por el microprocesador empujado para la implementación del algoritmo de comparación de huellas dactilares. La Figura 5.9 también muestra el reparto de tareas entre el microprocesador empujado y el coprocesador hardware. Como se aprecia en la Figura 5.9, el coprocesador realiza la transformada wavelet y el cálculo de CC, S y SS, mientras que el microprocesador empujado, Microblaze en este caso, se encarga del resto de las tareas, destacando entre ellas el envío de datos al coprocesador, la reconfiguración del mismo y la recomposición de los resultados obtenidos.

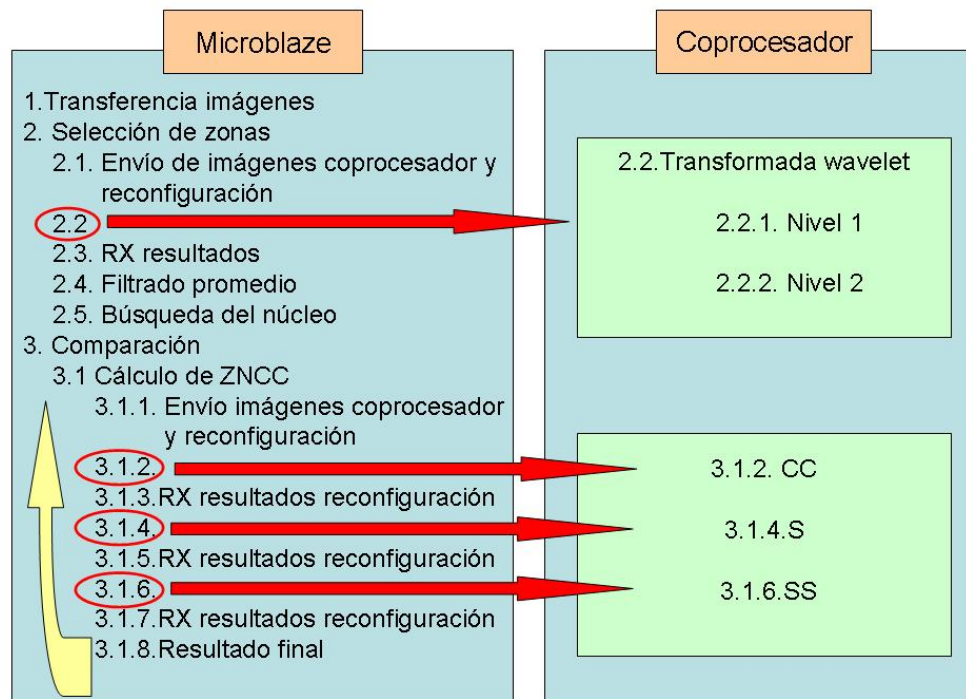


Figura 5.9. Esquema del código ejecutado por el microprocesador empujado y las partes realizadas por el coprocesador hardware

Tal y como muestra la Figura 5.9, la primera tarea que se realiza es el envío de las imágenes a comparar al microprocesador empotrado. Una vez recibidas se comienza con las tareas propias del algoritmo de comparación para imágenes de alta resolución descrito en el apartado 3.4 del capítulo 3. Este algoritmo se subdivide en dos grandes tareas: selección de zonas y comparación, tareas 2 y 3 respectivamente en el diagrama de la Figura 5.9. En la tarea 2, el microprocesador empotrado envía las imágenes al coprocesador hardware para que este realice la transformada wavelet de nivel 1 y 2, tareas 2.2.1 y 2.2.2 respectivamente en el diagrama de la Figura 5.9. Para realizar este proceso es necesario realizar sucesivas reconfiguraciones del coprocesador hardware. Una vez realizadas las tareas 2.2.1 y 2.2.2 el coprocesador envía los resultados al microprocesador empotrado, tarea 2.3, que realiza el resto de tareas necesarias para la selección de zonas (filtrado promedio, tarea 2.4, y búsqueda del núcleo, tarea 2.5). A partir de las zonas seleccionadas, el microprocesador empotrado realiza la comparación de las huellas gracias al cálculo de ZNCC. El coprocesador empotrado realiza el cálculo de CC, tarea 3.1.2, S, tarea 3.1.4 y SS, tarea 3.1.6. Es necesario que el microprocesador empotrado reconfigure el coprocesador y envíe los datos a procesar en cada una de las tareas a realizar por el coprocesador hardware. Una vez que el coprocesador ha finalizado la tarea a realizar enviará los resultados al microprocesador empotrado, que se encargará de recomponerlos y con ellos poder tomar una decisión sobre la similitud de las huellas dactilares comparadas.

5.7- Resultados experimentales

El SoPC diseñado se ha prototipado en una placa de evaluación de Xilinx (ML-506) [Xil_ML506], que contiene una FPGA Virtex 5 [Xil_V5] SX50T (XCVSX50T-1FFG1136).

En los siguientes subapartados se detallan los resultados obtenidos en términos de precisión, consumo de recursos y rendimiento. Con el objetivo de comparar las

prestaciones del SoPC diseñado con otras plataformas se ha realizado una versión software del algoritmo de verificación de huellas dactilares y se ha medido su rendimiento tanto en Microblaze sin coprocesador hardware como en un PC equiparable a un Pentium IV a 3.2 GHz con 2GB de RAM.

Los resultados experimentales presentados en este apartado se han obtenido para el SoPC con coprocesador con arquitectura dinámica. Los resultados con el coprocesador estático son equivalentes a los que se obtienen con un coprocesador dinámico de idéntico tamaño.

5.7.1- Precisión

Se han realizado medidas de precisión para cada uno de los modos de operación del coprocesador y para el algoritmo de verificación de huellas dactilares. A su vez se han realizado implementaciones software del algoritmo tanto en C como en Matlab para poder realizar medidas de precisión de los resultados. Las pruebas se han realizado con imágenes de huellas de la base de datos DB2 del FVC2006 [FVC2006], cuyo tamaño es de 560x400 píxeles y cada píxel representa un nivel de gris mediante 8 bits.

En los experimentos no se han apreciado diferencias significativas en precisión de los resultados obtenidos con el SoPC y con las implementaciones software. Ligeras diferencias en la precisión, menores del 0.07%, se han observado en los resultados de los diferentes modos de operación, pero estas diferencias son tan pequeñas que no afectan al resultado global del algoritmo. La causa de estas ligeras diferencias de precisión se debe principalmente a la diferencia en número de bits de los microprocesadores utilizados (Microblaze utiliza 32 bits y el PC tiene un microprocesador de 64 bits). Estas diferencias podrían reducirse utilizando un microprocesador empujado más potente. También podría aumentarse la precisión del coprocesador hasta los 48 bits, pero ya que el resto de los elementos del sistema

utiliza 32 bits, este cambio no repercutiría substancialmente en un aumento de la precisión.

5.7.2- Consumo de recursos

El consumo de recursos del sistema depende del tamaño de las imágenes y del tamaño de la matriz de DSP slices. Las Tablas 5.1 y 5.2 muestran los resultados para diferentes tamaños de imagen y de matriz de DSP slices. Los resultados de ambas tablas han sido obtenidos con ISE y EDK 9.1. La Tabla 5.1 muestra el consumo de recursos para el SoPC completo mientras que la Tabla 5.2 muestra el consumo de recursos para el coprocesador reconfigurable.

Matriz DSP slices	Tamaño Imágen	Slices	LUTs	DSP Slices	RAM Blocks
4x48	500x400	16457 (50%)	17917 (54%)	203 (70%)	32 (24%)
4x48	250x200	16452 (50%)	16884 (51%)	203 (70%)	32 (24%)
2x24	500x400	7619 (23%)	9980 (30%)	57 (19%)	32 (24%)
2x24	250x200	7613 (23%)	9658 (29%)	57 (19%)	32 (24%)

Tabla 5.1. Consumo de recursos del SoPC

El consumo de recursos del coprocesador reconfigurable se debe fundamentalmente al tamaño de la matriz de DSP slices y de las líneas de retardo.

Las Tablas 5.1 y 5.2 muestran que el consumo de recursos de esta arquitectura es moderado. En la Tabla 5.1 se puede observar que la mitad de la FPGA está vacía

y por tanto el SoPC admite más componentes.

Matriz DSP slices	Tamaño Imágen	Slices	LUTs	DSP Slices	RAM Blocks
4x48	500x400	10170 (31%)	11679 (35%)	196 (68%)	0 (0%)
4x48	250x200	10165 (31%)	10646 (32%)	196 (68%)	0 (0%)
2x24	500x400	1332 (4%)	3329 (10%)	50 (17%)	0 (0%)
2x24	250x200	1326 (4%)	3006 (9%)	50 (17%)	0 (0%)

Tabla 5.2. Consumo de recursos del coprocesador reconfigurable

La frecuencia máxima del SoPC es 100 MHz, que es la frecuencia de Microblaze y del OPB. El coprocesador se ha optimizado para esta frecuencia pero con la arquitectura descrita podría optimizarse hasta alcanzar los 250 MHz. La Tabla 5.3 muestra la frecuencia para el coprocesador, tal y como viene dada por la herramienta de síntesis, para distintos tamaños de imagen y de la matriz de DSP slices.

Matriz DSP slices	Tamaño Imágen	Máximo Retardo (ns)	Frecuencia (MHz)
4x48	500x400	8.53	117.10
4x48	250x200	8.53	117.13
2x24	500x400	7.44	134.35
2x24	250x200	7.25	138.01

Tabla 5.3. Máxima frecuencia del coprocesador reconfigurable

5.7.3- Rendimiento

El rendimiento se ha medido para el SoPC, PC y Microblaze sin coprocesador. Se han utilizado diferentes tamaños de imágenes, de filtros de wavelet y matrices de DSP slices.

La Tabla 5.4 muestra los resultados de rendimiento para el modo de operación transformada wavelet.

Matriz DSP slices	Tamaño Imagen	SoPC (ms)	Microblaze (ms)	PC (ms)
4x4	220x220	2.32	112.78	2.58
4x4	110x110	0.59	27.68	0.64
2x2	220x220	2.31	31.02	1.01
2x2	110x110	0.59	7.61	0.26

Tabla 5.4. Rendimiento del modo de operación transformada wavelet

Para este modo de operación el SoPC está trabajando mucho más rápido que Microblaze (hasta 49 veces más rápido) pero solo ligeramente más rápido que el PC. Los resultados experimentales muestran que la aceleración conseguida por el SoPC depende principalmente del tamaño de la matriz de DSP slices con una tendencia a aumentar la aceleración al aumentar el tamaño de la matriz.

Los dos tamaños de la matriz de DSP slices utilizados para los experimentos no requieren procesamiento de resultados parciales por parte del microprocesador empotrado, ya que ambos filtros bidimensionales caben completos en el coprocesador. Si la dimensión vertical fuera mayor de 4 (número máximo de columnas de la matriz de DSP slices disponibles en la FPGA utilizada), Microblaze

tendría que operar con los resultados parciales para conseguir el resultado final de cada filtrado. En tal caso, a pesar de la penalización introducida por el cálculo del resultado final mediante Microblaze, el rendimiento del SoPC mejoraría notablemente gracias a la aceleración conseguida por el coprocesador.

La Tabla 5.5 muestra el rendimiento para el modo de operación CC.

Tamaño T	Tamaño I	SoPC (ms)	Microblaze (ms)	PC (ms)
48x48	220x220	242	38047	684
48x48	110x110	33	5046	95
24x24	220x220	143	12405	223
24x24	110x110	28	2419	48

Tabla 5.5. Rendimiento del modo de operación CC

En este caso el SoPC tiene un rendimiento más elevado que los otros dos sistemas. En particular, si se consideran las imágenes de mayor tamaño, el SoPC es 157 veces más rápido que Microblaze y alrededor de 3 veces más rápido que el PC. Para los tamaños considerados el coprocesador está aprovechando los recursos de la matriz de DSP slices al máximo nivel y por eso se aumenta el rendimiento de manera considerable.

Estos resultados experimentales demuestran que un microprocesador empotrado de recursos limitados, como Microblaze a una frecuencia de 100 MHz, junto con un coprocesador hardware puede superar en prestaciones a un PC de última generación a 3 GHz.

Las Tablas 5.6 y 5.7 muestran el rendimiento de los modos de operación S y SS. En ambos modos los resultados son similares al modo de operación CC. Para los tres modos (CC, S, SS) el rendimiento del SoPC es el mismo, pero sin embargo el rendimiento de los otros dos sistemas para S y SS es mejor. Esto es debido a que, de los tres modos considerados, el que conlleva una mayor carga computacional es CC

y es por ello el que consigue una mejor aceleración gracias al coprocesador hardware.

En todos los experimentos realizados para los modos CC, S y SS, el coprocesador realiza cálculos parciales ya que en ninguno de los casos cabe la imagen completa dentro de la matriz de DSP slices, cuyo tamaño máximo para la FPGA utilizada es de 4x48. Por lo tanto, para todos los experimentos considerados para estos modos, Microblaze realiza el cálculo de los resultados finales a partir de los resultados parciales generados por el coprocesador.

Tamaño T	Tamaño I	SoPC (ms)	Microblaze (ms)	PC (ms)
48x48	220x220	242	19813	634
48x48	110x110	33	2627	87
24x24	220x220	143	6460	214
24x24	110x110	28	1260	42

Tabla 5.6. Rendimiento del modo de operación S

Tamaño T	Tamaño I	SoPC (ms)	Microblaze (ms)	PC (ms)
48x48	220x220	242	21591	658
48x48	110x110	33	2863	94
24x24	220x220	143	7034	222
24x24	110x110	28	1370	47

Tabla 5.7. Rendimiento del modo de operación SS

En las Tablas 5.5, 5.6 y 5.7 se aprecia que cuando el tamaño de la matriz de DSP slices se reduce a la mitad, la aceleración que consigue el SoPC no se reduce en la misma proporción. Esto es debido a que aunque se disminuya la matriz

también se disminuye el tamaño de la imagen y con ello el número de resultados parciales y el esfuerzo computacional que realiza Microblaze.

Además de realizar experimentos con cada modo por separado, se han realizado experimentos para obtener el rendimiento del algoritmo completo de verificación de huellas dactilares. Las imágenes utilizadas pertenecen a la base de datos DB2 del FVC2006 [FVC2006] con un tamaño de 560x400 píxeles con 8 bits por píxel. La zona seleccionada mediante la transformada wavelet tiene un tamaño de 150x150 píxeles.

La Tabla 5.8 muestra el rendimiento del algoritmo de verificación de huellas dactilares para los 3 sistemas: SoPC, Microblaze y PC. El algoritmo se ha dividido en dos etapas: selección de zonas (realizada con la transformada wavelet) y comparación (realizada con ZNCC).

	SoPC (ms)	Microblaze (ms)	PC (ms)
Selección de zona	123.01	883.89	54.7
Comparación	632.29	28548.77	834.4
Total	755.3	29432.66	889.1

Tabla 5.8. Rendimiento del algoritmo de verificación de huellas dactilares (Patrón de 48x48)

Para la primera etapa, el primer nivel de la transformada wavelet se realiza sobre una imagen de 560x400 píxeles y el segundo nivel sobre un cuarto del tamaño de esta imagen. En este algoritmo únicamente se realizan dos niveles de la transformada wavelet. La familia utilizada es Daubechies 2 con lo cual se necesitan 4 filtros bidimensionales de tamaño 4x4 para cada nivel de la transformada. Por lo tanto el tamaño de la matriz de DSP slices es de 4x4 para el modo de operación transformada wavelet. Con el resultado obtenido de esta etapa se selecciona una zona de 150x150 píxeles dentro de la imagen de la huella dactilar de entrada para pasar a la siguiente etapa.

En la etapa de comparación se realiza ZNCC de la zona seleccionada de la imagen de entrada con un patrón de 48x48. Para los modos de operación CC, S y SS el tamaño de la matriz de DSP slices es de 4x48. Al contrario que en el modo transformada wavelet, en estos modos es necesario que Microblaze elabore los resultados finales a partir de los parciales ya que no cabe la imagen entera dentro de la matriz de DSP slices.

Los resultados experimentales muestran que la implementación del algoritmo en el SoPC alcanza un rendimiento similar al de un PC de altas prestaciones. Cabe destacar los limitados recursos del microprocesador empujado y su reducida frecuencia de funcionamiento (100 MHz) en comparación con la del PC (3.2 GHz). Los resultados experimentales también demuestran la marcada aceleración que se consigue con el SoPC frente a la implementación en Microblaze debida a la utilización del coprocesador hardware reconfigurable. Si se deseara conseguir mejores prestaciones, podría recurrirse a microprocesadores más potentes como por ejemplo Power PC.

En la Tabla 5.8 también se muestra que el rendimiento del SoPC es mejor que el de un PC para realizar la etapa de comparación del algoritmo. Esto se debe a las altas prestaciones del coprocesador para realizar operaciones MAC de alta velocidad. Sin embargo, para la etapa de selección de zona el PC es el que consigue el mejor rendimiento. Esto es debido a que las operaciones que se realizan durante esta etapa requieren que Microblaze realice una búsqueda sobre la imagen inicial de entrada (560x400 píxeles) para determinar la zona seleccionada. Esta búsqueda penaliza considerablemente el rendimiento global de la etapa ya que la transformada wavelet de dos niveles solamente tarda en el SoPC 39.54 ms. Estas operaciones realizadas por Microblaze podrían ser optimizadas en gran medida relajando los criterios de búsqueda o incluso realizando parte de las mismas con el coprocesador. Se ha preferido que Microblaze realice la búsqueda frente a otras posibles soluciones para mostrar las penalizaciones que puede sufrir cualquier algoritmo real con el sistema diseñado. Estas penalizaciones son debidas a las bajas prestaciones del microprocesador empujado empleado. Sin embargo, aún incluyendo estas penalizaciones se puede concluir gracias a los resultados

experimentales que el rendimiento del algoritmo implementado en el SoPC está en el mismo orden de magnitud que un PC con un microprocesador de 3.2 GHz.

En la Tabla 5.9 se muestran los resultados obtenidos realizando el mismo experimento pero considerando un patrón de tamaño 24x24, es decir reduciendo el tamaño del patrón un cuarto. Como se aprecia en la Tabla 5.9 los resultados para la etapa de selección de zonas son iguales que los presentados en la Tabla 5.8, ya que el cambio de tamaño del patrón no afecta a esta etapa.

	SoPC (ms)	Microblaze (ms)	PC (ms)
Selección de zona	123.01	883.89	54.7
Comparación	727.18	11281.69	337.5
Total	850.19	12165.58	392.2

Tabla 5.9. Rendimiento del algoritmo de verificación de huellas dactilares (Patrón de 24x24)

Sin embargo, en la etapa de comparación se aprecia un incremento en el tiempo requerido por el SoPC para realizar esta tarea respecto al presentado en la Tabla 5.8. Este incremento se debe a que mientras el tamaño del patrón disminuye, se mantiene constante el tamaño de la imagen de entrada (150x150) y por tanto es necesario realizar un mayor número de desplazamientos. En definitiva, esto implica un mayor número de correlaciones que el requerido para el patrón 4 veces mayor. Este aumento del número de correlaciones repercute negativamente en el rendimiento del sistema, ya que se aumenta el número de resultados que debe recomponer el microprocesador empotrado.

En la Tabla 5.10 se muestran desglosadas las tareas para la etapa de comparación en el SoPC considerando tamaños de patrón de 24x24 píxeles y 48x48 píxeles.

Tarea	T (24x24) (ms)	T (48x48) (ms)
CC	71.94	100.25
S	60.08	87.50
SS	60.08	87.50
Otras operaciones	535.09	356.72

Tabla 5.10. Rendimiento de la etapa de comparación en el SoPC

La Tabla 5.10 muestra que aunque para el patrón de 48x48 el tiempo para cada uno de los modos (CC, S y SS) aumenta ligeramente, el resto de operaciones realizadas por el microprocesador empotrado requieren un menor tiempo que el necesario para un patrón de 24x24. Estos resultados demuestran que aunque el coprocesador hardware tiene un rendimiento muy alto en las tareas que desempeña, el microprocesador empotrado puede empeorar el rendimiento global del sistema debido a sus escasas prestaciones.

5.8- Conclusiones

En este capítulo se ha propuesto una arquitectura de SoPC de altas prestaciones para la aceleración de algoritmos de comparación de huellas dactilares basado en técnicas de correlación. Estos algoritmos utilizan operaciones típicas del campo de procesamiento de imágenes conllevando una alta complejidad computacional. El sistema diseñado puede utilizarse para implementaciones de otros algoritmos en el campo del procesamiento de imágenes.

De las dos soluciones propuestas, la más versátil es la arquitectura reconfigurable, por su adaptación al tamaño de los datos y a la operación realizada gracias a la reconfiguración dinámica. La reconfiguración la realiza directamente el

microprocesador a través del bus OPB reduciendo el tiempo de reconfiguración al mínimo y por lo tanto la reconfiguración no afecta al rendimiento de la arquitectura. La reconfiguración de grano grueso se realiza vía OPB en 4 transacciones de bus, sin necesidad de utilizar los mecanismos propios de la FPGA (regeneración y envío del bitstream). Es por tanto una solución óptima en tiempo de reconfiguración y en velocidad de procesado.

Cabe destacar del coprocesador diseñado la posibilidad de incluir nuevas operaciones sin modificar la arquitectura descrita. Es por tanto idóneo para su utilización en la implementación de algoritmos en el ámbito del procesado de imágenes.

De los resultados experimentales se concluye que el coprocesador es altamente eficiente para todos los modos de operación, superando en rendimiento a un PC de altas prestaciones. Considerando el algoritmo en su conjunto, las prestaciones del SoPC con un microprocesador empotrado a 100 MHz son equiparables a las de un PC de altas prestaciones (3.2 GHz y 2 GB de RAM). Esto demuestra que la arquitectura diseñada puede acelerar algoritmos implementados para SoPC igualando las prestaciones de un PC de gama alta.

La arquitectura diseñada, unida a una eficiente asignación de tareas, facilitaría enormemente las implementaciones de SoPC. Con una librería de coprocesadores reconfigurables se podrían alcanzar altas prestaciones para infinidad de algoritmos facilitando el nivel de abstracción del diseñador y acercándolo a los actuales compiladores de lenguajes de alto nivel.

Capítulo 6:

CONCLUSIONES

6.1-	CONCLUSIONES	166
6.2-	LÍNEAS FUTURAS	172

6.1- Conclusiones

Las técnicas de comparación de huellas dactilares basadas en correlación son técnicas precisas, como demuestran los resultados obtenidos en [FVC2002], [FVC2004], pero poco estudiadas y con problemas de rendimiento debido a la alta carga computacional que presentan. En esta tesis se ha profundizado en las técnicas de correlación de huellas dactilares, proponiendo nuevas soluciones, así como su aceleración hardware mediante arquitecturas eficientes.

Uno de los objetivos marcados para esta tesis era el estudio de los métodos de comparación de huellas dactilares basados en técnicas de correlación. En el capítulo 3 de esta tesis se han presentado dos algoritmos para la comparación de huellas dactilares basados en técnicas de correlación. Ambos algoritmos son aportación original de esta tesis. Los algoritmos presentan soluciones diferenciadas según la resolución que presenten las imágenes a comparar. Los métodos de alineamiento utilizados en los dos algoritmos también son aportación original de esta tesis.

En ambos algoritmos se han tenido en cuenta como puntos de partida dos premisas importantes. En primer lugar, reducir al mínimo el preprocesado para considerar toda la información disponible en la huella y evitar la introducción de artefactos. Los resultados obtenidos permiten afirmar que es posible obtener tasas de error aceptables mediante técnicas de correlación sin necesidad de hacer un preprocesado exhaustivo de las imágenes de las huellas dactilares. No obstante, los resultados pueden ser mejorables si las imágenes tienen baja calidad o excesiva distorsión. En estos casos es preciso un refinamiento de los métodos propuestos o una combinación con otros métodos.

En segundo lugar, se ha tenido en cuenta en todo momento la necesidad de obtener prestaciones aceptables para permitir el uso de los métodos propuestos en aplicaciones en tiempo real. Para ello, y considerando la perspectiva de la aceleración hardware de los algoritmos, se han propuesto técnicas que presentan una alta regularidad y similitud con las técnicas de correlación, de manera que su

implementación hardware pueda ser eficiente. En esta línea, se han propuesto técnicas de alineamiento originales que se basan en la correlación del campo de orientación, para imágenes de baja resolución, y en el uso de la transformada wavelet, para imágenes de alta resolución. Ambas técnicas pueden implementarse en hardware de forma similar y compatible con la correlación, permitiendo un eficiente uso de los recursos. Gracias a estas características, es posible soportar la mayoría de los cálculos más pesados en un mismo coprocesador con unas necesidades de reconfiguración mínimas, como se demuestra en el capítulo 5.

El algoritmo para imágenes de baja resolución tiene cuatro etapas: preprocesado, alineamiento, selección de zonas y comparación. La etapa de preprocesado se encarga de adecuar la calidad de la imagen a la requerida por las etapas posteriores. A partir de la imagen preprocesada, en la etapa de alineamiento se determinan el desplazamiento y rotación entre las imágenes. Como subproducto de esta etapa se obtiene el campo de orientación de las imágenes. El campo de orientación es utilizado por la etapa de alineamiento para determinar el desplazamiento y rotación entre las imágenes. Para ello se realiza la correlación cruzada de las orientaciones del campo de orientación de ambas huellas. En la etapa de selección de zonas se decide qué partes de las imágenes van a ser comparadas. La selección de zonas se realiza para reducir la carga computacional, ya que se evita comparar la imagen entera, y además para hacer frente al efecto global provocado por la distorsión no lineal. En esta etapa, las zonas que se eligen como óptimas son zonas distintivas y con buena calidad de imagen. La selección de zonas se realiza gracias a la coherencia de la orientación de las imágenes, así como al solape entre ambas imágenes. El solape entre ambas imágenes se determina gracias a la rotación y el desplazamiento determinados previamente en la etapa de alineamiento. Una vez seleccionadas las zonas de ambas imágenes (área de búsqueda y zonas candidatas) se realiza la comparación mediante el cálculo de ZNCC de las zonas seleccionadas.

El algoritmo para imágenes de baja resolución se ha probado con la base de datos de concurso FVC2000 DB2 A [FVC2000] con 800 huellas de tamaño 256x364 píxeles. El EER obtenido es del 8%. Analizando los errores de comparación obtenidos, se encuentran dos causas básicas para los fallos: baja

calidad de las imágenes y escaso solapamiento entre muestras. Ambos tipos de error podrían solucionarse con etapas adicionales de preprocesado para determinar tanto la calidad de la imagen como el área de solape y rechazar aquellas muestras que no cumplan unos requisitos preestablecidos.

El algoritmo para imágenes de alta resolución tiene dos etapas: selección de zonas y comparación. En la primera etapa de este algoritmo se localiza la posición del núcleo en las imágenes para poder así seleccionar una zona alrededor de este punto. La localización del núcleo se realiza mediante la transformada wavelet de las imágenes, formando gracias a ella el campo de orientación predominante (H/V) que muestra en cada píxel de la imagen la dirección predominante entre las direcciones vertical y horizontal. El campo de orientación predominante (H/V) muestra un patrón triangular para todas las huellas independientemente de su tipo y los efectos que sufra la muestra (distorsión, presión, rotación, desplazamiento, etc.). Este patrón presenta siempre uno de sus vértices apuntando a la posición del núcleo. Por lo tanto la localización del núcleo se realiza mediante la determinación de ese vértice del patrón triangular. Una vez localizada la posición del núcleo se determina una zona de tamaño fijo alrededor del mismo para la etapa de comparación. Seleccionadas las zonas, en la etapa de comparación se calcula ZNCC de estas zonas. En este algoritmo, la zona candidata se subdivide en varias partes y se correla cada una de ellas de forma independiente con el área de búsqueda, para hacer frente al efecto de la distorsión no lineal sobre la comparación.

El algoritmo para imágenes de alta resolución se ha probado con la base de datos de concurso FVC2006 DB2 [FVC2006] con 120 huellas de tamaño 560x400 píxeles. El EER obtenido fue del 9.865%. El método propuesto para la detección del núcleo ha producido sólo 4 errores en esta base de datos, fallando por tanto en el 3.3% de las huellas. Analizando los errores de comparación obtenidos, se encuentran tres causas básicas para los fallos: la calidad de las huellas, núcleos cercanos al extremo de la imagen y baja curvatura en las huellas. Las vías para minimizar los efectos de estos errores y mejorar con ello la precisión se han detallado en el capítulo 3.

Ambos algoritmos presentan buenos resultados en cuanto a precisión que

podrían ser mejorables por diversas vías. El rendimiento de los algoritmos en un PC de gama alta requiere más de un segundo en ambos casos y por tanto estos algoritmos son candidatos ideales para su aceleración hardware. Cabe destacar que las implementaciones concretas que se han realizado de los algoritmos utilizan un número limitado de rotaciones y desplazamientos de las zonas a correlar, por tanto un cálculo más exhaustivo requeriría una mayor carga computacional con la consiguiente disminución del rendimiento de los algoritmos. Además de esto, cabe destacar también que los algoritmos de comparación de huellas dactilares pueden estar integrados en sistemas portables y por tanto cuanto menor coste, potencia y espacio requiera el sistema, mayores beneficios se podrán obtener del producto final.

De los resultados de rendimiento de los algoritmos presentados en el capítulo 3 se extrae la conclusión que la mayor parte de la carga computacional es debida al cálculo de ZNCC. A la luz de los resultados obtenidos en el capítulo 3, en el capítulo 4 se han presentado arquitecturas hardware para la aceleración del cálculo de la correlación. Todas las arquitecturas presentadas son aportación original de esta tesis.

La arquitectura espacial presentada se basa en la utilización de una matriz sistólica de DSP slices para la realización del cálculo de la correlación cruzada. Cada DSP slice de la matriz realiza una MAC y pasa el resultado al siguiente DSP slice de su misma fila. Por lo tanto, en cada fila de la matriz se calcula la correlación cruzada de una fila de ambas imágenes. La arquitectura también necesita líneas de retardo para pasar los resultados de una fila a la siguiente en el instante temporal correcto.

La arquitectura espectral presentada se basa en la utilización de FFT slices (FFT unidimensional de una fila de una imagen). En esta arquitectura se busca realizar el máximo número de FFTs en paralelo. En este caso, el manejo y acceso a las memorias es complejo debido a la gran cantidad de FFTs requeridas para el cálculo.

Para la normalización se presentan arquitecturas recursivas para el cálculo de S y SS que comparten con las otras arquitecturas el flujo de datos de entrada. Con esta

aproximación es posible combinar ambos circuitos para integrarlo en un solo chip. Además de esto, al compartir el flujo de datos los resultados se pueden generar en paralelo, sin disminuir el rendimiento.

De los resultados experimentales presentados en el capítulo 4 se puede concluir que las arquitecturas presentadas pueden mejorar el rendimiento para el cálculo de la correlación hasta en dos órdenes de magnitud, sin pérdida de precisión, con respecto a una implementación software en un PC de gama alta. Estos resultados se han obtenido con una tecnología, Virtex 4, que hoy en día ya ha sido superada. Cabe esperar que las ventajas se mantengan o incluso mejoren con el progreso incesante de la tecnología FPGA.

Un resultado interesante de esta tesis es que la solución espectral, basada en el teorema de la correlación, no proporciona ventajas significativas respecto a la solución espacial en su implementación hardware. Esta es una diferencia significativa con respecto a las soluciones software, donde es conocido que la solución espectral es más eficiente porque reduce el número de cálculos mediante el uso de la FFT. Sin embargo, en implementaciones hardware la solución espectral requiere un elevado uso de recursos que inciden negativamente en la frecuencia de operación y en el número de unidades básicas que pueden operar en paralelo, mientras que la solución espacial es más sencilla arquitecturalmente y se adapta mucho mejor a los recursos hardware disponibles en una FPGA.

Una vez analizadas las arquitecturas hardware para la aceleración del cálculo de la correlación, se han propuesto soluciones a nivel de sistema para poder integrar las arquitecturas hardware presentadas en el capítulo 4 en un SoPC. En el capítulo 5 de esta tesis se ha presentado una arquitectura básica de SoPC compuesta por un microprocesador empotrado más un coprocesador hardware. En esta arquitectura, el coprocesador se encarga de las tareas de mayor carga computacional para mejorar el rendimiento del sistema. Se han propuesto dos arquitecturas para el coprocesador del SoPC: coprocesador estático y coprocesador dinámico. Ambas arquitecturas son aportación original de esta tesis. Los diseños presentados en este capítulo se han implementado para la tecnología concreta de un fabricante de FPGAs, aunque cabe destacar que bajo mínimas modificaciones serían portables a otras tecnologías.

La arquitectura estática se basa en la arquitectura espacial presentada en el capítulo 4, pero conteniendo una única fila de DSP slices, además de memorias para almacenamiento de resultados intermedios y un sistema de control para el flujo de datos. En esta arquitectura, el coprocesador se encarga de realizar las tareas de mayor carga computacional (CC, S, SS) y el microprocesador el resto de tareas necesarias para la comparación, incluyendo el envío de datos al coprocesador y la recomposición de resultados.

La arquitectura dinámica se compone de un coprocesador reconfigurable de grano grueso que puede cambiar en tiempo de ejecución tanto la operación a realizar como el tamaño de las imágenes empleadas, implicando un redimensionamiento de la matriz de DSP slices del coprocesador. La reconfiguración la realiza directamente el microprocesador empotrado, sin necesidad de utilizar los mecanismos internos de reconfiguración de la FPGA.

Este coprocesador presenta una solución versátil a la par que eficiente. La pérdida de posibilidades frente a una reconfiguración de grano fino se ve compensada por una disminución drástica del tiempo de reconfiguración, que con la arquitectura propuesta puede realizarse con tan solo cuatro transacciones de 32 bits.

El sistema se ha prototipado para una FPGA Virtex 5, utilizando como microprocesador empotrado Microblaze. Los resultados experimentales presentados en el capítulo 5 muestran que el SoPC diseñado es altamente eficiente para diversas operaciones comunes en el campo del procesamiento de imágenes (CC, S, SS y transformada wavelet) superando en rendimiento a implementaciones software en un PC de altas prestaciones. En el SoPC propuesto también se ha implementado el algoritmo de comparación para imágenes de alta resolución presentado en el capítulo 3 de esta tesis consiguiendo el mismo rendimiento que un PC de altas prestaciones.

A la luz de los resultados obtenidos con esta tesis se puede afirmar que las técnicas de correlación son técnicas prometedoras en el campo de la comparación de huellas dactilares, tanto por su precisión como por su capacidad de comparar toda la información contenida en la huella. En el campo de la algorítmica con esta

tesis se han abierto numerosas vías que podrán ser perfeccionadas con trabajos futuros.

El problema del rendimiento de estas técnicas también se ha solventado eficientemente tal y como muestran los resultados experimentales aportados en los capítulos 4 y 5 de esta tesis. Las arquitecturas propuestas hacen posible cálculos de correlación en tiempo real y en sistemas de bajo coste y tamaño. El SoPC con coprocesador dinámico propuesto presenta una solución muy versátil, ya que es capaz de modificar en tiempo de ejecución la tarea a realizar por el coprocesador, el tamaño de las imágenes y de la matriz de DSP slices, realizando el cálculo de forma eficiente para un conjunto de datos de gran tamaño.

6.2- Líneas futuras

A partir del trabajo realizado durante esta tesis, surgen futuras líneas de trabajo que se resumen a continuación.

En primer lugar, en cuanto a los algoritmos de comparación de huellas dactilares basados en técnicas de correlación, un campo interesante es la combinación de las técnicas propuestas con otros métodos, con el fin de mejorar las tasas de error. Como se ha indicado anteriormente, una de las premisas de este trabajo era minimizar el preprocesado y utilizar al máximo la información disponible en las imágenes originales de las huellas. No obstante, si las huellas son de muy mala calidad, el beneficio que produce el preprocesado compensa el riesgo de introducir efectos indeseados.

En segundo lugar, como todas las demás técnicas de comparación de huellas dactilares, las técnicas propuestas pueden ser refinadas mediante un ajuste óptimo de los parámetros de los algoritmos y la introducción de correcciones específicas para algunos tipos particulares de huellas. Para ello es necesario realizar una experimentación exhaustiva, con diferentes bases de datos de huellas, provenientes

de diferentes tipos de sensores, diferentes resoluciones, etc. Esta experimentación formaría parte del desarrollo de herramientas de uso comercial y por tanto queda fuera del propósito de esta tesis.

Finalmente, es importante destacar que las técnicas propuestas son extrapolables al ámbito general del procesamiento de imágenes. En esta línea se abren numerosas posibilidades, puesto que la correlación, el filtrado y la convolución de imágenes figuran entre las técnicas más comunes en muchas aplicaciones.

Por ejemplo, las técnicas presentadas para detección de núcleo en el algoritmo para imágenes de alta resolución o el alineamiento para el algoritmo de baja resolución, podrían utilizarse en otros ámbitos facilitando la extracción de información relevante en otros tipos de imágenes. Esto también es aplicable a las arquitecturas presentadas en los capítulos 4 y 5, ya que se pueden utilizar para acelerar numerosos algoritmos en el campo del procesamiento de imágenes. La arquitectura del coprocesador hardware reconfigurable propuesta para SoPC es un claro ejemplo de la generalidad del trabajo presentado, ya que las operaciones a realizar pueden ser modificadas en tiempo de ejecución y además se facilita la realización de la misma operación sobre una cantidad elevada de datos de entrada (operación típica en el procesamiento de imágenes).

BIBLIOGRAFÍA

A

[Altera] <http://www.altera.com>

B

[Bariamis_04] Bariamis, D.G.; Iakovidis, D.K.; Maroulis, D.E.; Karkanis, S.A.:
“An FPGA-based architecture for real time image feature extraction”, Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004, 23-26 Aug. 2004, vol. 1, pp: 801 - 804.

[Barrenechea_07] Barrenechea, M.; Altuna, J.; San Miguel, M.: “A Low-Cost FPGA-based Embedded Fingerprint Verification and Matching System”, 2007 Fifth Workshop on Intelligent Solutions in Embedded Systems, 21-22 June 2007, pp: 250 – 261.

[Bazen_00] Bazen, A. M., Verwaaijen, G. T. B., Gerez, S. H., Veelenturf, L. P. J., van der Zwaag, B. J.: "A Correlation-Based Fingerprint

Verification System”. Proceedings Workshop on Circuits System and Signal Processing (ProRISC 2000), 2000, pp: 205 – 213.

[Bazen_02] Bazen, A.M.; Gerez, S.H.: “Systematic methods for the computation of the directional fields and singular points of fingerprints”, IEEE Transactions on Pattern Analysis and Machine Intelligence, July 2002, vol. 24, Issue 7, pp: 905–919.

[Becker_07] Becker, J., Hubner, M., Hettich, G., Constapel, R., Eisenmann, J., Luka, J.: “Dynamic and Partial FPGA exploitation”, Proceedings of the IEEE, Feb. 2007, vol. 95, No 2, pp: 438-452.

[Biometrika] http://www.biometrika.it/eng/prodotti_comp.html

[Bowen_08] Bowen, O., Bouganis, C.S.: “Real-time image super resolution using an FPGA”, International Conference on Field Programmable Logic and Applications, 2008, FPL '08, pp: 89-94.

[Brunelli_08] Brunelli, C., Garzia, F., Nurmi, J., Campi, F., Picard, D.: “Reconfigurable hardware: the holy grail of matching performance with programming productivity“, International Conference on Field Programmable Logic and Applications, 2008, FPL '08, pp: 409-414.

C

[Chen_94] Qin-Sheng Chen; Defrise, M.; Deconinck, F., “Symmetric phase-only matched filtering of Fourier-Mellin transforms for image

registration and recognition”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 16, Issue 12, Dic. 1994, pp:1156 – 1168.

[Choi_06] Jae Seung Choi; Jin Kyu Lim; Jae Yeal Nam; Yeong Ho Ha: “Image capture and storage system for digital TV without degrading image quality”, IEEE Transactions on Consumer Electronics, Mayo 2006, vol. 52, Issue 2, pp:467 – 471.

[Coetzee_93] Coetzee, L., Botha, E. C.: “Fingerprint Recognition in Low Quality Images”. Pattern recognition, 1993, vol. 26, no 10, pp: 1441 – 1460.

[Crouzil_96] Crouzil, A., Massip-Pailhes, L., Castan, S.: “A new correlation criterion based on gradient fields similarity”. Proceedings of the 13th International Conference on Pattern Recognition, 25-29 Agosto 1996, vol. 1, pp: 632 – 636

D

[DeCastro_87] De Castro, E., Morandi, C.: “Registration of translated and rotated images using finite Fourier Transforms”. IEEE Transactions on Pattern Analysis and Machine Intelligence. Sept. 1987, PAMI-9, 5, , pp: 700-703.

[Derrode_01] Derrode, S., Ghorbel, F.: “Robust and efficient Fourier-Mellin transform approximations for Gray-level image reconstruction and complete invariant description”. Computer Vision and Image Understanding, Julio 2001, vol. 83 no. 1, pp: 57-78.

- [Diaz_06] Diaz, J.; Ros, E.; Mota, S.; Pelayo, F.; Ortigosa, E.M.: “Subpixel motion computing architecture”, IEE Proceedings -Vision, Image and Signal Processing, Dic. 2006, Volume 153, Issue 6, pp:869 – 880.
- [Dillinger_06] Dillinger, P.; Vogelbruch, J.F.; Leinen, J.; Suslov, S.; Patzak, R.; Winkler, H.; Schwan, K.: “FPGA-Based Real-Time Image Segmentation for Medical Systems and Data Processing”, IEEE Transactions on Nuclear Science, Agosto 2006, Volume 53, Issue 4, Part 2, pp: 2097 – 2101.

E

- [Elbirt_00] Elbirt, A. J., Paar, C.: “An FPGA implementation and performance evaluation of the serpent block cipher. ACM/SIGDA”. Internacional Symposium on FPGAs, 2000, pp 33–40.
- [Entrena_04] Entrena, L., Sanchez Reillo, R., Lindoso, A., Liu, J.: “Fingerprint matching acceleration with smartcards”, Design of Systems and Integrated Circuits 04, DCIS 04, Noviembre 2004.

F

- [FVC2000] Maio, D., Maltoni, D., Cappelli, R., Wayman, J.L., Jain, A.K.: “FVC2000: Fingerprint Verification Competition”. IEEE Transactions on Pattern Analysis Machine Intelligence, 2002, vol 24(3), pp: 402–412.

- [FVC2002] Maio, D., Maltoni, D., Cappelli, R., Wayman, J.L., Jain, A.K.: “FVC 2002: Second Fingerprint Verification Competition”. 16th Proc. Int. Conf. on Pattern Recognition, 2002, vol. 3, pp: 811-814.
- [FVC2004] Cappelli, R.; Maio, D.; Maltoni, D.; Wayman, J.L.; Jain, A.K.: “Performance evaluation of fingerprint verification systems”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Enero 2006, vol 28, Issue 1, pp: 3 – 18.
- [FVC2006] <http://bias.csr.unibo.it/fvc2006/databases.asp>.
- [Fons_07] Fons, F., Fons, M., Canto, E.: “Approaching Fingerprint Image Enhancement through Reconfigurable Hardware Accelerators”, IEEE International Symposium on Intelligent Signal Processing, 2007. WISP 2007. 3-5 Oct. 2007, pp: 1 – 6.
- [FX3000] <http://www.biometrika.it/eng/fx3000.html>

G

- [Germain_97] Germain, R.S., Califano, A., Colville, S.: “Fingerprint matching using transformation parameter clustering”, Computational Science and Engineering, IEEE, Octubre.-Diciembre 1997, vol. 4, Issue 4, pp: 42 – 49.
- [Gonzales_92] Gonzales, R. C., Woods, R.E.: “Digital Image Processing”, Addison-Wesley, Reading, MA, 1992.

[Gonzales_08] Gonzales, R. C., Woods, R. E.: “Digital Image Processing”, Pearson Prentice Hall, 2008.

[Gupta_06] Gupta, A.K., Nooshabadi, S., Taubman, D., Dyer, M.: “Realizing Low-Cost High-Throughput General-Purpose Block Encoder for JPEG2000”, IEEE Transactions on Circuits and Systems for Video Technology, Julio 2006, Volume 16, Issue 7, pp: 843 – 858.

H

[Hartenstein_00] Hartenstein, R., Hoffmann, Th., Nageldinger, U.: “Design-Space Exploration of Low Power Coarse Grained Reconfigurable Datapath Array Architectures”, Lecture Notes in Computer Science, Integrated Circuit Design, LNCS 1918, 2000, Springer-verlag Berlin Heidelberg, pp: 118-128.

[Hatami_05] Hatami, S., Hosseini, R., Kamarei, M., Ahmadi, H.: “Wavelet based fingerprint image enhancement”, IEEE International Symposium on Circuits and Systems, 2005. ISCAS 2005, vol. 5, pp: 4610 - 4613.

[Hatano_02] Hatano, T., Adachi, T., Shigematsu, S., Morimura, H., Onishi, S., Okazaki, Y., Kyuragi, H.: “A fingerprint verification algorithm using the differential matching rate”, Proceedings of the 16th International Conference on Pattern Recognition, 2002, vol. 3, pp: 799–802.

- [Hong_98] Lin Hong, Yifei Wan, Jain, A.: “Fingerprint image enhancement: algorithm and performance evaluation”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1998, vol 20, Issue 8, pp: 777 – 789.
- [Huang_00] Huang, W. J., Saxena, N., Mccluskey, E. J.: “A reliable LZ data compressor on reconfigurable coprocessors”, IEEE Symposium on Field-Programmable Custom Computing Machines, 2000, pp: 249–258.
- [Huang_04] Huang, K., Aviyente, S.: “Choosing best basis in wavelet packets for fingerprint matching”, International Conference on Image Processing, 2004, ICIP '04, 24-27 Octubre 2004, vol 2, pp:1249 – 1252.
- [Huvenandana_00] Huvenandana, S., Changick Kim, Jenq-Neng Hwang: “Reliable and fast fingerprint identification for security applications”, International Conference on Image Processing, 2000, Proceedings, 10-13 Septiembre 2000, vol. 2, pp:503 – 506.

I

J

- [Jain_97] Jain, A. K., Hong, L., Pankanti, S., Bolle, R.: “An Identity Authentication System using Fingerprints”, Proceedings of the IEEE, sep. 1997, vol. 85, Issue 9, pp: 1365-1388.

- [Jain_99] Jain, A., Bolle, R., Pankanti, S.: “Biometrics: Personal identification in a networked society”, Kuwler Academia Publishers, 1999.
- [Jain_00] Jain, A. K., Prabhakar, S., Hong, L., Pankanti, S.: “Filterbank-Based Fingerprint Matching”, IEEE Transactions on Image Processing, 2000, vol. 9, pp: 846 – 859.
- [Jain_01] Jain, A., Ross, A., Prabhakar, S.: “Fingerprint matching using minutiae and texture features”, International Conference on Image Processing, 2001, Proceedings, Octubre 2001, vol. 3, pp: 282 – 285.
- [Jain_06] Jain, A., Chen, Y., Demetrius, M.: “Pores and Ridges: Fingerprint Matching using Level 3 Features”, Proc. 18th Int. Conf. on Pattern Recognition (ICPR’06), 2006, vol. 4, pp: 477-480.
- [Jain_L_99] Jain, L.C., Halici, U., Hayashi, I., Lee, S. B., Tsutsui, S.: “Intelligent biometric techniques in fingerprint and face recognition”, CRC Press LLC, 1999, pp: 11-45.
- [Jean_99] Jean, J.S.N., Tomko, K., Yavagal, V., Shah, J., Cook, R.: “Dynamic reconfiguration to support concurrent applications”, IEEE Transactions on Computers, Junio 1999, vol. 48, Issue 6, pp: 591 – 602.
- [Jiang_07] Jiang Li, Tulyakov, S., Govindaraju, V.: “Verifying Fingerprint Match by Local Correlation Methods”, First IEEE International

Conference on Biometrics: Theory, Applications, and Systems, 2007. BTAS 2007, 27-29 Sept. 2007, pp: 1 – 5.

- [Jiang_M_07] Jiang, M.; Crookes, D.: “Area-efficient high-speed 3D DW processor architecture”, *Electronics Letters*, Abril 2007, vol. 43, Issue 9, pp:502 – 503.

K

- [Kitaoka_03] Toshiro Kitaoka, Hideharu Amano, Kenichiro Anjo: “Reducing the Configuration Loading Time of a Coarse Grain Multicontext Reconfigurable Device”, *Lecture Notes in Computer Science* 2778, *Field-Programmable Logic and Applications*, 2003, Springer-verlag Berlin Heidelberg, pp: 171-180.

- [Klima_97] Klima, M.; Rott, J.; Dvorak, P.; Gleeson, D.; McKenna-Lawlor, S.; Keating, J., “Model of 2D optical correlator for fingerprint identification”, *IEEE Aerospace and Electronic Systems Magazine*, Julio 1997, vol. 12, Issue 7, pp: 3 – 9.

- [Kovacs_00] Kovacs-Vajna, Z. M.: “A Fingerprint Verification System Based on Triangular Matching and Dynamic Time Warping”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000, vol 22, pp: 1266 – 1276.

L

- [Lee_02] Dongjae Lee, Kyoungtaek Choi, Jaihie Kim: “A robust fingerprint matching algorithm using local alignment”, 16th

International Conference on Pattern Recognition, Agosto 2002, Proceedings, vol. 3, pp:803 – 806.

- [Liang_03] Xuejun Liang, Jean, J.S.-N.: “Mapping of generalized template matching onto reconfigurable computers”, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Junio 2003, vol 11, Issue 3, pp: 485 – 498.
- [Lifeng_04] Sha Lifeng, Tang Xiaou: “Orientation-improved minutiae for fingerprint matching”, Proceedings of the 17th International Conference on Pattern Recognition, ICPR 2004, Agosto 2004, vol. 4, pp:432 – 435.
- [Lim_04] Lim, E., Toh, K. A., Suganthan, P.N., Jiang, X., Yan, W.Y.: “Fingerprint quality analysis”, Proceedings on International Conference on Image Processing, IEEE, 2004, pp: 1241-1244.
- [Lindoso_05] Lindoso, A., Entrena ,L., López-Ongil, C., Liu, J.: “Correlation-based fingerprint matching using FPGAs”, IEEE 2005 Conference on Field-Programmable Technology, FPT 05, Diciembre 2005.
- [Lindoso_06] Lindoso, A., Entrena, L., San-Millan, E., Garcia-Lorenz, M.: “High Performance automatic fingerprint recognition”, Design of Systems and Integrated Circuits 06, DCIS 06, Noviembre 2006.

- [Lindoso_07_a] Lindoso, A., Entrena, L., Izquierdo, J.: “FPGA-Based acceleration of fingerprint minutiae matching”, IEEE Southern conference on Programmable Logic 07, SPL 07, Febrero 2007.
- [Lindoso_07_b] Lindoso, A., Entrena, L., Liu-Jimenez, J., San Millan, E.: “Correlation-based Fingerprint Matching with Orientation Field Alignment” , 2nd Internacional Conference on Biometrics, ICB 07, LNCS, Springer, Agosto 2007.
- [Lindoso_07_c] Lindoso, A., Entrena, L., Liu-Jimenez, J.: “Wavelet-based fingerprint region selection”, 12th Internacional Conference on Computer analysis of Images and Patterns, CAIP 07, LNCS, Springer, Agosto 2007.
- [Lindoso_07_d] Lindoso, A., Entrena, L., Liu-Jimenez, J., San Millan, E.: “Increasing security with correlation-based fingerprint matching”, IEEE International Carnahan Conference on Security Technology, ICST 2007, Octubre 2007.
- [Lindoso_07_e] Lindoso, A., Entrena, L.: “High Performance FPGA-based image correlation”, Journal of Real Time Image Processing, Dic 2007, Vol. 2, Ed. Springer-Verlag, pp: 223-233.
- [Lindoso_08] Lindoso, A., Entrena, L., Izquierdo, J., Liu-Jimenez, J.: “Coarse-grain dynamically reconfigurable coprocessor for image processing in SoPC”, IEEE International conference on Field Programmable Logic and Applications, FPL 2008, Septiembre 2008.

- [Lopez_06] López García, M.L.; Cantó Navarro, E.F.: “FPGA Implementation of a Ridge Extraction Fingerprint Algorithm Based on Microblaze and Hardware Coprocessor”, International Conference on Field Programmable Logic and Applications, 2006, FPL '06, Agosto 2006, pp: 1 – 5.

M

- [Maio_97] Maio, D., Maltoni, D.: “Direct gray-scale minutiae detection in fingerprints”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1997, vol. 19, Issue 1, pp: 27–40.
- [Mallat_98] Mallat, S.G.: “A wavelet tour of signal Processing”, Academic Press, 1998.
- [Maltoni_03] Maltoni, D., Maio, D., A. K. Jain., A.K., Prabhakar, S.: “Handbook of Fingerprint Recognition”. Springer-Verlag New York, 2003.
- [McDonald_08] McDonald, E.J.: “Runtime FPGA partial reconfiguration”, IEEE Aerospace and Electronic Systems Magazine, Julio 2008, vol. 23, Issue 7, pp: 10 – 15.
- [Memik_03] Memik, S.O., Katsaggelos, A.K., Sarrafzadeh, M.: “Analysis and FPGA implementation of image restoration under resource constraints”, IEEE Transactions on Computers, Marzo 2003, vol. 52, Issue 3, pp: 390 – 399.

- [Miao_05] Miao-li Wen, Yan Liang, Quan Pan, Hong-cai Zhang: “A Gabor filter based fingerprint enhancement algorithm in wavelet domain”, 2005 IEEE International Symposium on Communications and Information Technology, ISCIT 2005, pp: 1468 - 1471
- [Mokju_04] Mokju, M., Abu-Bakar, S.A.R.: “Fingerprint matching based on directional image constructed using expanded Haar wavelet transform”, International Conference on Computer Graphics, Imaging and Visualization, 2004, CGIV 2004, Proceedings, 26-29 Julio 2004, pp: 149 – 152.
- [Moon_05] Moon, Y.S., Chen, J.S., Chan, K.C., So, K., Woo, K.C.: “Wavelet based fingerprint liveness detection”, Electronics Letters, 29 Sept. 2005, vol. 41, Issue 20, pp: 1112 – 1113.

N

- [Nguyen_06] Nguyen, D., Halupka, D., Aarabi, P., Sheikholeslami, A.: “Real-time face detection and lip feature extraction using field-programmable gate arrays”, IEEE Transactions on Systems, Man, and Cybernetics, Agosto 2006, Part B, vol. 36, Issue 4, pp: 902 – 912.

O

- [opencv] <http://www.intel.com/technology/computing/opencv/>

- [Ouyang_06] Zhengyu Ouyang, Jianjiang Feng, Fei Su, Anni Cai: “Fingerprint Matching With Rotation-Descriptor Texture Features”, 18th International Conference on Pattern Recognition, 2006, ICPR 2006, vol. 4, pp: 417 – 420.

P

- [Park_03] Park, C. H., Oh ,S. K., Kwak, D. M., Kim, B. S., Song, Y. C., Park, K. H.: “A new reference point detection algorithm based on orientation patron labelling in fingerprint images”, Lecture Notes in Computer Science (LNCS), vol. 2652, 2003, pp: 697-703.
- [Parziale_06] Parziale, G., Diaz-Santana, E., Hauke, R.: “The Surround Imager: A Multicamera Touchless Device to Acquire 3D Rolled-Equivalent Fingerprints”, Proceedings of International Conference on Biometrics, Lecture Notes in Computer Science, LNCS, vol. 3832, 2006, pp: 244-250.

Q

R

- [Ratha_96] Ratha, N.K., Karu, K., Shaoyun Chen, Jain, A.K.: “A real-time matching system for large fingerprint databases”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Agosto 1996, vol. 18, Issue 8, pp: 799 – 813.

S

- [Saegusa_08] Saegusa, T., Maruyama, T., Yamaguchi, Y.: “How fast is an FPGA in image processing?”, International Conference on Field Programmable Logic and Applications, 2008, FPL '08, pp: 77-82.
- [Sedcole_06] Sedcole, P., Blodget, B., Becker, T., Anderson, J., Lysaght, P.: “Modular dynamic reconfiguration in Virtex FPGAs”, IEE Proceedings-Computers and Digital Techniques, 2 Mayo 2006, vol. 153, Issue 3, pp: 157 – 164.
- [Seow_02] Seow, B.C., Yeoh, S.K., Lai, S.L., Abu, N.A.: “Image based fingerprint verification”, Student Conference on Research and Development, 2002, SCORd 2002, 16-17 Julio 2002, pp: 58 – 61.
- [Siegel_82] Siegel, L. J., Siegel, H. J., Feather, A.E.: “Parallel Processing Approaches to Image Correlation”, IEEE Transactions on Computers, Marzo 1982, vol. C-31, no. 3, pp: 208-218.
- [Shuckers_04] Shuckers, S., Abhyankar, A.: “Detecting liveness in fingerprint scanners using wavelets: Results of the test Dataset”, Biometric Authentication, Book Series Lecture Notes in Computer Science, volume 3087/2004, Springer-Verlag Berlin Heidelberg, 2004.
- [Stosz_94] Stosz, J. D., Alyea, L. A.: “Automated System for fingerprint Authentication using Pores and Ridge Structure”, Proceedings of

the SPIE (Automatic Systems for the identification and inspection of Humans), 1994, vol 2277, pp: 210 – 223.

T

- [Tan_08] Heng Tan; DeMara, R.F.: “A Multilayer Framework Supporting Autonomous Run-Time Partial Reconfiguration”, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Mayo 2008, vol. 16, Issue 5, pp: 504 – 516.
- [Tico_01_a] Tico, M., Immonen, E., Ramo, P., Kuosmanen, P., Saarinen, J.: “Fingerprint recognition using wavelet features”, The 2001 IEEE International Symposium on Circuits and Systems, 2001. ISCAS 2001, vol. 2, pp: 21 - 24.
- [Tico_01_b] Tico, M.; Kuosmanen, P.; Saarinen, J.: “Wavelet domain features for fingerprint recognition”, Electronics Letters, Enero 2001, vol. 37, Issue 1, pp:21 – 22.
- [Tico_03] Tico, M., Kuosmanen, P.: “Fingerprint matching using an orientation-based minutia descriptor”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Agosto. 2003, vol. 25, Issue 8, pp: 1009 – 1014.
- [Tong_04] Xi-Feng Tong, Xiang-Long Tang, Jian-Hua Huang, Xiao Li: “Fingerprint minutiae matching based on complex minutiae vector”, International Conference on Machine Learning and Cybernetics, 2004, Proceedings of 2004, Agosto 2004, vol. 6, pp: 3731 – 3735.

U

- [USP_02] “Systems and methods with identity verification by streamlined comparison and interpretation of fingerprints and the like”. US Patent 6356649. 12 Marzo, 2002.
- [Uzun_05] Uzun, I.S., Amira, A., Bouridane, A.: “FPGA implementations of fast Fourier transforms for real-time signal and image processing”, IEE Proceedings-Vision, Image and Signal Processing, Junio 2005, vol. 152, Issue 3, pp: 283 – 296.

V

- [Vancourt_05] VanCourt, T., Yongfeng Gu, Herbordt, M.C.: “Three-dimensional template correlation: object recognition in 3D voxel data”, Seventh International Workshop on Computer Architecture for Machine Perception, 2005, CAMP 2005, 4-6 Julio 2005, pp: 153 – 158.
- [Venkataramani_03] Krithika Venkataramani, B.V.K. Vijaya Kumar: “Fingerprint Verification Using Correlation Filters”, Lecture Notes in Computer Science, Audio- and Video-Based Biometric Person Authentication, Springer Berlin / Heidelberg, 2003, Volume 2688/2003, pp: 886-894.

W

- [Walker_99] Walker, J. S.: “A primer on wavelets and their scientific applications”, Ed. CRC Press LLC, 1999.

- [Wayman_05] Wayman, J., Jain, A., Maltoni, D., Maio, D.: “Biometric Systems, Technology, Design and Performance Evaluation” Springer- Verlag London, 2005.
- [Wei_02] Wei-Peng Zhang, Qing-Ren Wang, Tang, Y.Y.: “A wavelet-based method for fingerprint image enhancement”, 2002 International Conference on Machine Learning and Cybernetics, 2002, vol. 4, pp: 1973 – 1977.
- [Willis_01] Willis, A. J., Myers, L.: “A Cost-Effective Fingerprint Recognition System for Use with Low-Quality Prints and Damaged fingerprints”, Pattern Recognition, 2001, vol. 34, no. 2, pp: 255 – 270.
- [Wilson_97] Wilson, C.L., Watson, C. I., Paek, E. G., “Combined Optical and Neural Network Fingerprint Matching”, Proceedings of SPIE (Optical Pattern Recognition VIII), 1997, vol. 3073, pp: 373-382.
- [Wilson_00] Wilson, C.L., Watson, C. I., Paek, E. G.: “Effect of resolution and image quality on combined optical and neural network fingerprint matching”, Pattern recognition, Elsevier, 2000, vol. 33, pp: 317-331.

X

- [Xiao_86] Xiao, Q., Bian, Z.: “An approach to Fingerprint Identification by Using the Attributes of Feature Lines of Fingerprint”, Proceedings of the International Conference on Pattern Recognition, 1986, pp: 663 – 665.

- [Xilinx] <http://www.xilinx.com>
- [Xil_BRAM] “Block RAM (BRAM) Block (v1.00a)”, www.xilinx.com, Xilinx Inc., 2004.
- [Xil_DSPV4] “XtremeDSP for Virtex-4 FPGAs User Guide”, www.xilinx.com, Xilinx Inc., 2007.
- [Xil_DSPV5] “Virtex-5 FPGA XtremeDSP design considerations”, www.xilinx.com, Xilinx Inc., Febrero 2008.
- [Xil_FSL] “Fast Simple Link (FSL) Bus v.2.11a”, www.xilinx.com, Xilinx Inc., Junio 2007.
- [Xil_FFT] “Fast Fourier Transform v3.0”. www.xilinx.com, Xilinx LogiCore, Xilinx Inc., 2004.
- [Xil_mblaze] “Microblaze Processor Reference Guide”, www.xilinx.com, Xilinx Inc., Junio 2007.
- [Xil_ML506] “ML505/506 Evaluation Platform”, www.xilinx.com, Xilinx Inc., Octubre 2007.
- [Xil_LMB] “Local Memory Bus (LMB) v1.0 (v.1.00a)”, www.xilinx.com, Xilinx Inc., Abril 2005.

- [Xil_OPB] “On-chip peripheral bus v.2.0 with OPB arbiter (v.1.10c)”, www.xilinx.com, Xilinx Inc., Diciembre 2005.
- [Xil_pblaze] “Picoblaze 8-bit Embedded microcontroller User Guide (v.1.1.2)”, www.xilinx.com, Xilinx Inc., Junio 2008.
- [Xil_PLB] “Processor Local Bus (PLB) v4.6 (v1.00a)”, www.xilinx.com, Xilinx Inc., Agosto 2007.
- [Xil_PPC] “PowerPC Processor Reference Guide (v1.2)”, www.xilinx.com, Xilinx Inc., Enero 2007.
- [Xil_Spartan3] “Spartan-3A DSP FPGA Family: Data Sheet”, www.xilinx.com, Xilinx Inc., Junio 2008.
- [Xil_V4] “Virtex-4 Family Overview”. www.xilinx.com, Xilinx Inc., 2004.
- [Xil_V5] “Virtex-5 Family Overview”, www.xilinx.com, Xilinx Inc.
- [Xinge_04] Xinge You, Jianwei Yang, Yuan Yan Tang, Bin Fang, Luoqing Li: “Skeletonization of Fingerprint Based-on Modulus Minima of Wavelet Transform”, *Advances in Biometric Person Authentication*, Book Series Lecture Notes in Computer Science, 2004, vol. 3338/2004, Springer- Verlag Berlin Heidelberg.

- [Xudong_00] Jiang Xudong, Yau Wei-Yun: "Fingerprint minutiae matching based on the local and global structures", 15th International Conference on Pattern Recognition, Septiembre 2000, vol. 2, pp: 1038-1041.

Y

- [Yahagi_90] Yahagi, H., Igaki, S., Yamagishi, F.: "Moving-Window Algorithm For Fast Verification", Proceedings Conference Southeastcon, 1990, pp: 343 – 348.

- [Yik_04] Yik-Hing Fung, Yuk-Hee Chan: "Fingerprint recognition with improved wavelet domain features", Proceedings of 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing, 2004, pp: 33 – 36.

- [Yoo_07] Yoo, Jang-Hee, Ko, Jong-Gook, Chung, Yun-Su, Jung, Sung-Uk, Kim, Ki-Hyun, Moon, Ki-Young, Chung, Kyoil: "Design of Embedded Multimodal Biometric Systems", 2007 Third International IEEE Conference on Signal-Image Technologies and Internet-Based System, 16-18 Dic. 2007, pp: 1058 – 1062.

Z

- [Zhang_06] Jianxin Zhang, Zongying Ou, Honglei Wei: "Fingerprint Matching Using Phase-Only Correlation and Fourier-Mellin Transforms", Sixth International Conference on Intelligent Systems Design and Applications, 2006. ISDA '06, Oct. 2006, vol. 2, pp: 379 – 383.

ACRÓNIMOS

AFIS	Automatic Fingerprint Identification System
ASIC	Application Specific Integrated Circuit
BRAM	Block RAM
CC	Correlación cruzada
DMA	Direct Memory Access
DSP	Digital Signal Processor
EER	Equal Error Rate
FAR	False Acceptance Rate
FFT	Fast Fourier Transform
FIFO	First In First Out
FMR	False Match Rate
FNMR	False Non Match Rate
FPGA	Field Programmable Gate Array
FRR	False Rejection Rate
FSL	Fast Simple Link
FVC	Fingerprint Verification Competition
LMB	Local Memory Bus

LUT	Look-Up Table
MAC	Multiply ACcumulation
OPB	On-chip Peripheral Bus
PCI	Peripheral Component Interconnect
PLB	Processor Local Bus
RAM	Random Access Memory
RISC	Reduced Instruction Set Computer
ROC	Receiver Operating Characteristic curve
S	Suma de píxeles de una imagen
SoC	System On Chip
SoPC	System on Programmable Chip
SS	Suma del cuadrado de los píxeles de una imagen
TIC	Tecnologías de la información y comunicación
USB	Universal Serial Bus
VHDL	VHSIC Hardware Description Language
XMD	Xilinx Microprocessor Debugger
ZNCC	Zero-mean Normalized Cross-Correlation
